Università degli Studi di Udine

Dipartimento Politecnico di Ingegneria ed Architettura

Dottorato di Ricerca in Ingegneria Industriale e dell'Informazione

Ph.D. Thesis

Simulation Tools and Developments on Integral Formulations for the Computation of Eddy Currents

Candidate

Mauro Passarotto

Supervisor

Prof. Ruben Specogna

Reviewers

Prof. Christophe Geuzaine

Prof. Antonello Tamburrino

Cycle XXXII — Year 2020

Institute Contacts
Dipartimento Politecnico di Ingegneria ed Architettura
Università degli Studi di Udine
Via delle Scienze, 206
33100 Udine — Italia
https://dpia.uniud.it/

Author's Contacts
mauro.passarotto90@gmail.com

To all my teachers.

Acknowledgements

In this page, I would like to reserve the first thanks to my supervisor, Prof. Ruben Specogna, for the inexplicable trust he had in me. Also, I thank him for the uncountable time he devoted to my research and to all our countless projects; I would not have been able to conclude this Ph.D. without his incredibly varied knowledge both theoretical and practical. For sure, I will always consider him as a model for his patience and his enthusiasm in pursuing new goals. Then, I would like to thank Prof. Christophe Geuzaine for the too brief period I have been spending at Université de Liège in the Applied Computational Electromagnetics group. I am really grateful to him for the time we spent discussing about applied mathematics and for answering to all my often stupid questions without losing his patience. Despite being undoubtedly busy, he has always been able to find some time for me in his so-popular agenda. Furthermore, I wish to express my gratitude to all the people of the ACE lab too, for their kind hospitality and for letting me think I am able to speak a decent French, even if not true. Finally, I here thank Prof. Antonello Tamburrino and Prof. Christophe Geuzaine (again) for having thoroughly reviewed this thesis and for their advice that has surely improved the content quality and clearness.

During these three years, I had the chance of sharing the office with many brilliant Ph.D. students; amongst the others, I want to mention Bernard Kapidani for tutoring me since when I arrived in Udine, Michele Libralato because he always knows a tool that can do what you need to do, and Francesco Stacchi for our common fate and the sarcastic discussions we had during this doctorate that helped me surviving this experience.

I thank my whole family for supporting me during all the many "Degrees" of my life and for their sacrifice that I have been appreciating more and more when growing up. I am also really grateful for helping me living out of a suitcase without ever burdening me because of my absence.

And, thank you Maria. Because of all my discouraged moments that you had to bear and because of the distance that we have been managing for all these years. You are the last of this page but you have always been the first one in my mind and in my heart; and you will always be.

This work has been supported by the project "HEaD HIGHER EDUCATION AND DEVELOPMENT OPERAZIONE 1 – UNIUD" - Regione autonoma Friuli-Venezia Giulia - Fondo Sociale Europeo - Programma operativo 2014/2020

Abstract

Computational electromagnetics is a discipline that since many years ago has permitted deep innovations in the study of electromagnetic problems. Even if, nowadays, commercial software undeniably shows a certain maturity when applied to practical problems, some research work has still to be done in going beyond the theoretical limits underneath the various approaches.

With respect to this, integral formulations still present some open issues. Historically, the exploitation of these formulations to study eddy currents started around the 90s with the seminal works of G. Albanese, R. Martone and R. Rubinacci together with the research activity of L. Kettunen and L. R. Turner and then with G. Meunier, who more recently rediscovered them. Lately, the contributions of L. Codecasa, R. Specogna and F. Trevisan have further increased the possibilities offered by this approach by introducing a set of new shape functions for polyhedral grids that are based on a discrete geometrical reinterpretation of the physics of electromagnetic phenomena.

One of the main features characterizing integral formulations to compute eddy currents stems from the fact that they do not require any discretization of the complement of the conductor to be studied. As a drawback, they lead to fully populated matrices whose assembly results to be remarkably time consuming and whose size can sometimes saturate the memory of the calculator. In this respect, this composition presents a new volume integral code for polyhedral grids describing how a fast and efficient cohomology computation can be implemented to treat also non-simply connected domains. Then, some tools are provided for the reduction of the size, and thus of the assembly time too, of the fully populated matrix. More precisely, the attention is focused on the exploitation of cyclic symmetry and on the novel topology-related issues arising when integral formulations have to be referred only to the symmetry cell of the complete conducting domain in order not to spoil the block-circulant property of the system matrix when building the cohomology generators or the gauging tree. Furthermore, also new iterative methods are considered as additional approaches to limit the size of the system matrix to be assembled: despite being already known to the computational electromagnetics community, their convergence behaviour has not been studied yet when they are applied to integral formulations as the one here proposed. Specifically, after presenting a purely iterative scheme derived from the volume integral formulation whose convergence can be somehow problematic, we propose a new direct-iterative method based on Krylov subspace techniques and on the domain splitting into multiple conductors that exhibits a much improved behaviour. The study of these methods leads to new interesting findings to be considered in addition to matrix compression techniques.

Contents

List of Tables

List of Figures

1

Introduction

Integral formulations applied to electromagnetic problems have become popular since many years ago especially for the study of full wave models like, amongst others methods, the partial elements equivalent circuit (PEEC) proposed in 1974 by A. E. Ruehli [1] or like the integral methods used to study electromagnetic scattering. The exploitation of such formulations to study eddy currents subsequently started around the 90s with the seminal works of R. Rubinacci, R. Martone and G. Albanese [2, 3] by whom the contributions of this thesis are inspired and, simultaneously, with the research activity of L. Kettunen and L. R. Turner [4]. Finally, more recently, the topic has been rediscovered for instance by G. Meunier [5], among others.

On one hand, these formulations have become appealing for the solution of eddy current problems because they do not require any discretization of the complement of the conductor wherein induced currents flow. This means that, beyond substantially reducing the number of elements to be considered when solving the problem, the whole process of the mesh construction can be focused on the conducting part only, a feature that results to be fundamental when the case of study exhibits a particularly complex geometry. Indeed, a typical setting where integral codes for eddy currents are widely employed is represented by the conducting shielding structure of fusion reactors whose design is undoubtedly intricate due to the presence of numerous holes and apertures. In addition, also the study of printed circuit boards (PCB) can somehow reach a similar level of complexity: usually, PEEC codes have always been used for such a purpose, but also a more general integral code could fulfil the same task if the working frequency does not exceed the limit within which eddy currents approximation holds.

On the other hand, reasons for which integral formulations applied to eddy current problems are still an open issue in the computational electromagnetics community can be identified with the main weakness of these methods: due to the integral constitutive relation used to relate the magnetic vector potential to the current density, the matrix expressing the problem results to be a dense matrix whose number of entries increases with the square of the degrees of freedom (DOFs) of the problem. This unpleasant feature leads to have troubles when the matrix has to be assembled and stored and it could also lead to memory overflows when computing the solution.

As a consequence, researchers have tried to overcome these difficulties by both improving the behaviour of the solvers and by reducing the size of the matrix. As far as the algorithms to solve the system are concerned, it is possible to state that efficient solvers are nowadays available given that several routines exist as the LAPACK routines of the Intel® MKL library. Similarly, also the problem of matrix storage has been successfully faced by resorting initially to fast multipole methods as in [6, 7] and then by means of effective compression techniques like the adaptive cross approximation (ACA) [8, 9] and the use of hierarchical matrix algebra [10]. These innovative techniques in addition to the incredibly fast improvements of the calculators performance allow to solve problems up to $20k$ DOFs without any compression, whereas in past limitations of the calculators memory imposed to deal with a number of DOFs of one order of magnitude smaller.

However, if on one hand the issue of the system solution does not represent a problem anymore, on the other hand the proposal of new techniques that can produce further steps forward in the direction of limiting the problem size is an aspect on which some work can still be done.

First, in this thesis we propose a novel volume integral formulation to study eddy currents on non-magnetic solid conductors whose mesh can be constituted by arbitrary polyhedra. This approach will be developed in order to deal with arbitrarily shaped *manifold* domains thanks to fast and efficient tools for the necessary pre processing for the *cohomology* computation here associated to polyhedral grids for the first time. Then, we treat the topological aspects arising when dealing with cyclic symmetry applied to integral formulations defined on a discrete geometrical framework with nodal or edge unknowns: in this regard, we will initially consider eddy current problems that can be studied thanks to a surface integral formulation to successively extend the main ideas employed in the 2D case to the volume integral code for polyhedra.

As final contribution, we reconsider iterative methods as additional tools that can be used when dealing with very large problems whose number of entries can be further reduced after the exploitation of matrix compression techniques by means of iterative schemes. This type of approach allows to alternatively express some parts of the dense matrix of the eddy current problem that by this way do not have to be stored but that can just be computed on the fly. As a consequent drawback, this choice leads to an iterative scheme whose convergence can be sometimes problematic and it has to be carefully studied, as in this thesis it is done.

Structure

This work is organized as follows. Initially, we first introduce the complete set of Maxwell's equations in which we will identify the correct subset representing eddy current problems; then, we start from the basic geometrical elements constituting a discretization grid of a given physical domain to subsequently construct an algebraic geometrical framework on which Maxwell's equation are recast to be expressed as geometrical entities. This will lead to Tonti's diagram on which an eddy current problem can be depicted as successive operations between matrices and arrays representing the global physical quantities and the discrete vector calculus operators of the electromagnetic problem. Later, after introducing the discrete counterpart of electric and magnetic constitutive laws within a Discrete Geometrical Approach, we outline the equations governing the volume integral formulation for non magnetic conducting domains. In regard

to this, we will continue the dissertation about this volume integral formulation by exploiting the benefits of domains exhibiting cyclic symmetry and by explaining which new topological issues can arise in this case. Finally, we move on and we will discuss about two new possible approaches to solve the same eddy current problem in an integral form by exploring the possibilities offered by iterative schemes. Indeed, we show that despite introducing some convergence-related issues that can be mitigated by means of Krylov subspace techniques, an iterative approach allows a non-negligible saving in terms of required memory for the system matrix assembly and storage. Eventually, the conclusion is drawn.

Author's contributions

The topics treated in this thesis have led to the following papers:

- P. Bettini, M. Passarotto, R. Specogna, "A volume integral formulation for solving eddy current problems on polyhedral meshes," *IEEE Trans. Magn.,* vol. 53, no. 6, June 2017;
- T. Maceina, P. Bettini, G. Manduchi, M. Passarotto, "Fast and Efficient Algorithms for Computational Electromagnetics on GPU Architecture," *IEEE Trans. on Nuclear Science,* vol. 64, no. 7, pp. 1983–1987, July 2017;
- P. Bettini, M. Passarotto, R. Specogna, "Coupling volume and surface integral formulations for eddy current problems on general meshes," *IEEE Trans. Magn.,* vol. 54, no. 3, March 2018;
- P. Bettini, M. Passarotto, R. Specogna, "Iterative solution of eddy current problems on polyhedral meshes," *IEEE Trans. Magn.,* vol. 54, no. 3, March 2018;
- M. Passarotto, R. Specogna, F. Trevisan, "Novel Geometrically Defined Mass Matrices for Tetrahedral Meshes," *IEEE Trans. Magn.,* vol. 55, no. 6, June 2019;
- B. Kapidani, M. Passarotto, R. Specogna, "Exploiting Cyclic Symmetry in Stream Function-Based Boundary Integral Formulations," *IEEE Trans. Magn.,* vol. 55, no. 6, June 2019;
- M. Passarotto, R. Specogna, "Cyclic symmetry in volume integral formulations for eddy currents: cohomology computation and gauging," *IEEE Trans. Magn.,* vol. 56, no. 2, February 2020;
- M. Passarotto, R. Specogna, C. Geuzaine "Fast iterative schemes for the solution of eddy current problems featuring multiple conductors by integral formulations," *IEEE Trans. Magn.,* DOI: 10.1109/TMAG.2019.2954071.

Also, this thesis has been inspired by them.

2

From Maxwell's equations to eddy currents

Electromagnetic (EM) phenomena are mathematically described by Maxwell's equations. After being published in 1862, during the time, their formalism and interpretation have been adapted to fit different purposes especially because of the advent of calculators. In this chapter we will first outline these equations in the usual vector calculus form to subsequently describe the eddy current problem since the main contributions of this work concern its solution.

2.1 Computational electromagnetics

Computational electromagnetics (CEM) is a discipline that, starting from computer science, physics and mathematics tries to develop efficient methods and tools to provide an approximate solution of Maxwell's equations describing a physical problem as a mathematical model. Despite identifying an exact year in which an EM was first solved by using a calculator is not meaningful, following the reasoning of [11], it is possible to assert that the first roots can be found around 1940 when Southwell exploited finite differences (FD) as first approach for the solution of a limited set of Maxwell's equations followed by G. Kron [12] that in 1944 established a method to represent Maxwell's equations with equivalent circuits. Since then many steps have been done, powered and driven by the continuous development in the digital computing that, especially after 1960s took place in the industrialized countries. Indeed, it was in the mechanical aircraft industry that finite elements method (FEM) was initially set up to then spread into other branches of physics like EM computation: in 1964 the seminal work of Winslow firstly introduced a FEM-like approach into EM by using an irregular triangular grid instead of the mapped squared grid used in FD and by applying the variational principle for the solution of magnetostatics.

Without continuing too long with this historical review, we can instead list the main features characterizing any computational method since the very beginning: a proper

choice of the Maxwell's equations to be considered for a given situation, an effective space discretization to describe the computational domain i.e. the finite portion of the 3D space where we are interested in computing EM fields and a valid mathematical framework that can represent the physics of the problem to be solved as a system of equations up to a certain acceptable error. In the initial chapters of this thesis, each of these three aspects will be treated.

2.2 Maxwell's equations

Maxwell's equations, written in a Cartesian frame and enforced in every regular point of $\mathbb{R}^3$, read

$$\nabla \times \boldsymbol{h}(\boldsymbol{r},t) - \partial_t \boldsymbol{d}(\boldsymbol{r},t) = \boldsymbol{j}_c(\boldsymbol{r},t), \tag{2.1a}$$
$$\nabla \cdot \boldsymbol{d}(\boldsymbol{r},t) = \rho_c(\boldsymbol{r}), \tag{2.1b}$$
$$\nabla \times \boldsymbol{e}(\boldsymbol{r},t) + \partial_t \boldsymbol{b}(\boldsymbol{r},t) = \boldsymbol{0}, \tag{2.1c}$$
$$\nabla \cdot \boldsymbol{b}(\boldsymbol{r},t) = 0. \tag{2.1d}$$

where, given that $\boldsymbol{r} \in \mathbb{R}^3$ represents a point in the 3D space, t a time instant and ∂_t the time derivative, the other symbols stand for, respectively:

- $\boldsymbol{b}$ and $\boldsymbol{h}$ the magnetic induction and the magnetic field;
- $\boldsymbol{d}$ and $\boldsymbol{e}$ the electric displacement and the electric field;
- $\boldsymbol{j}_c$ the conduction current field;
- ρ_c the free charge density.

These four equalities that, as it will be explained in the following sections, can be classified as *source* equations (the former two) and *configuration* equations (the remaining) are not sufficient to completely describe the EM field behaviour because some constitutive relations are required that express the connection between fields arising when a medium is considered. These relations can be written as

$$\boldsymbol{b}(\boldsymbol{r},t) = \mu(\boldsymbol{r})\boldsymbol{h}(\boldsymbol{r},t), \tag{2.2a}$$
$$\boldsymbol{j}_c(\boldsymbol{r},t) = \sigma(\boldsymbol{r})\boldsymbol{e}(\boldsymbol{r},t), \tag{2.2b}$$
$$\boldsymbol{d}(\boldsymbol{r},t) = \varepsilon(\boldsymbol{r})\boldsymbol{e}(\boldsymbol{r},t). \tag{2.2c}$$

where the newly introduced symbols are tensors representing the magnetic permeability μ, the electric conductivity σ and the dielectric permittivity ε. It is also possible to write the specular relations

$$\boldsymbol{h}(\boldsymbol{r},t) = \nu(\boldsymbol{r})\boldsymbol{b}(\boldsymbol{r},t), \tag{2.3a}$$
$$\boldsymbol{e}(\boldsymbol{r},t) = \rho(\boldsymbol{r})\boldsymbol{j}_c(\boldsymbol{r},t), \tag{2.3b}$$
$$\boldsymbol{e}(\boldsymbol{r},t) = \eta(\boldsymbol{r})\boldsymbol{d}(\boldsymbol{r},t) \tag{2.3c}$$

where clearly $\nu = \mu^{-1}$, $\rho = \sigma^{-1}$ and $\eta = \varepsilon^{-1}$.

2.2.1 Magnetodynamic

Selecting, approximating and recasting the right subset of (2.1),(2.2) and (2.3) is paramount to obtain a robust model. In fact, under certain conditions depending on the EM problem that has to be studied, we can ignore certain terms of (2.1) and select just some constitutive relations. The general setting of the eddy current problems here considered consists on a bounded domain $\Omega \in \mathbb{R}^3$ whose boundary is denoted with $\partial\Omega$ in which we can distinguish a conducting domain $\Omega_c \subset \Omega$, a source region $\Omega_s \subset \Omega$ with $\Omega_c \cap \Omega_s = 0$ and a non conducting or insulating region $\Omega_i = \Omega \setminus \{\Omega_c \cup \Omega_s\}$. Being interested in finding eddy currents in Ω_c, it is possible to neglect the term $\partial_t \boldsymbol{d}(\boldsymbol{r}, t)$ both in Ω_c and Ω_i. In fact, in the conducting region the displacement current is always negligible with respect to the conduction current $\boldsymbol{j}_c$ whereas in Ω_i the same approximation holds as long as the energy of electric field is a small fraction of the total energy. Consequently, in *magnetodynamic*, the equations to be considered to solve electromagnetics in Ω become

$$\nabla \times \boldsymbol{h}(\boldsymbol{r}, t) = \boldsymbol{j}_c(\boldsymbol{r}, t), \tag{2.4a}$$

$$\nabla \times \boldsymbol{e}(\boldsymbol{r}, t) + \partial_t \boldsymbol{b}(\boldsymbol{r}, t) = \boldsymbol{0}, \tag{2.4b}$$

$$\nabla \cdot \boldsymbol{b}(\boldsymbol{r}, t) = 0 \tag{2.4c}$$

with the constitutive relations $\boldsymbol{b}(\boldsymbol{r}, t) = \mu(\boldsymbol{r})\boldsymbol{h}(\boldsymbol{r}, t)$ and $\boldsymbol{e}(\boldsymbol{r}, t) = \rho(\boldsymbol{r})\boldsymbol{j}_c(\boldsymbol{r}, t)$. In addition, from equation (2.4a) it follows that

$$\nabla \cdot \boldsymbol{j}_c = 0 \tag{2.5}$$

i.e. the current density in the conductor must be solenoidal.

Theoretically, this should be the moment of substituting one equation into another, introduce some potentials (scalar and vector) and then obtain the typical *curl–curl* equation based, for instance, on the A–V formulation. Yet, another path will be followed in order to introduce some algebraic and geometrical concepts first and then represent the same equations in a more physical way.

3

Cell Method

After introducing Maxwell's equation describing EM fields, we will now focus on the representation of the 3D space. In fact, as already mentioned, to describe a real problem as a mathematical model, the computational domain has to be represented as a set of simpler geometrical elements. Usually, this is done by using triangles or polygons for 2D geometries whereas tetrahedra or polyhedra are used for a 3D space.

Different ways of conceiving these geometrical entities have been developed in literature, nevertheless we will here follow the reasoning of E. Tonti in [13] based on *cell complexes* and *algebraic topology* because, in a sense that will be more clear in the following, this allows to preserve in an exact way some important properties of EM fields in the computational model.

3.1 Cell complex

The starting point of the innovative approach proposed by Tonti and the spirit of this chapter can be summarized using the title of Tonti's paper: "*Why starting from differential equations for computational physics?*" [14]. Indeed, instead of adapting the computational domain representation to the vectorial calculus form of (2.4) which leads to the approximate solution of a partial differential equations (PDEs) system, we will first provide a consistent geometrical framework on which it is then possible to reformulate the eddy current problem in its very discrete form. To do this, we should start from the following definitions:

Definition 1. *A topological space is a space S together with a collection of open subsets T, called* ***topology*** *of the space that satisfies the conditions:*

1. $\emptyset \subset T$, *where* $\emptyset$ *is the empty set;*
2. S *is in* T*;*
3. *the intersection of any finite number of sets in* T *is also in* T*;*
4. *the union of an arbitrary number of sets in* T *is also in* T*.*

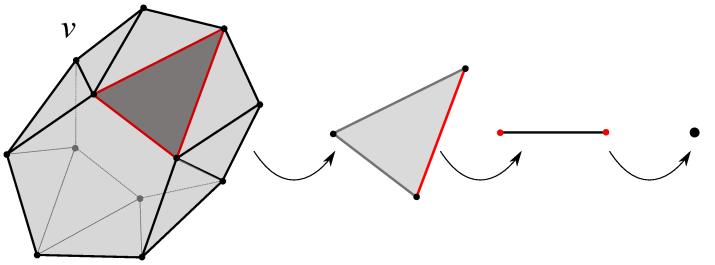

Figure 3.1: From the left to the right: a 3-cell, a 2-cell, a 1-cell, a 0-cell of a polyhedron v.

Definition 2. *The elements of a space S are called generically points and the sets $U \in T$ are called open sets. A set is open if $\forall x \in U$, x is surrounded only by elements of U. A closed set is a set whose complement is open.*

Definition 3. *A cell complex $\mathcal{K}$ is a collection of geometrical elements originating from a given subdivision of a space region. These elements are called cells.*

This last definition in particular perfectly fits the idea of FEM *mesh* or any discretization grid in general.

From the given definition, it follows that a cell complex of a 3D space $\Omega \in \mathbb{R}^3$ will be constituted by its n nodes, e edges, f faces and v volumes. Each of these elements can be classified as a p-cell σ_i^p where p is the dimension of the topological space which a cell belongs to; respectively, nodes are 0-cells because they are 0-dimensional, edges are 1-cells (1-dimensional), faces are 2-cells (2-dimensional) and finally, volumes are 3-cells (3-dimensional). A more meticulous description of p-cells can be given by introducing the concept of *homeomorphism* which by the way needs some additional definitions:

Definition 4. *An homomorphism is a map between two algebraic structures of the same type that preserves all the relevant structure.*

Definition 5. *An isomorphism is an homomorphism whose inverse exists and it is an homomorphism too.*

We are now ready to define the concept we were initially interested in:

Definition 6. *An homeomorphism is an isomorphism between topological spaces that respects their topological properties. It is a continuous and bijective map which has also a continuous inverse.*

Thanks to this, it is thus possible to state that a 3-cell is an homeomorphism of a ball, a 2-cell is an homeomorphism of a disk, a 1-cell is an homeomorphism of a segment. A 0-cell is just a dimensionless point, as already said.

Finally, a cell complex has also to satisfy the following property:

Property. *The intersection between two cells of a $\mathcal{K}$ complex is either a lower dimensional cell of $\mathcal{K}$ or it is empty.*

Consequences of this condition are exemplified in figure 3.2.

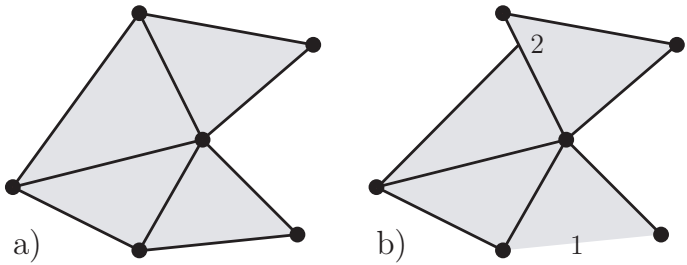

Figure 3.2: a) A conformal cell complex $\mathcal{K}$; b) a set of simplices that is not a cell complex: in 1 an edge is missing, in 2 the intersection between two simplices does not belong to $\mathcal{K}$.

Cell's orientation

Once that a cell complex has been defined, in order to build a geometrical framework that can describe physical laws, we have to introduce the concept of *orientation*. In fact, for instance, if we want to represent a flux through a surface $\mathcal{S}$ we need to assign a sign i.e. an orientation to that surface. More precisely, for our purpose, we say that any cell of a cell complex has to be endowed with either an *inner* or an *outer* orientation.

Broadly speaking, we specify that orienting a geometrical element means to consider the sub-elements composing it and then establish an order between them. For instance, given a segment it is possible to identify its endpoints and then decide which is the preceding and which is the following between the two.

Definition 7. *A p-cell in $\mathbb{R}^3$ is **inner** oriented when its orientation can be deduced by considering just points that belongs to the p-cell itself.*

Definition 8. *A p-cell in $\mathbb{R}^3$ is **outer** oriented when its orientation is deduced by considering points that are located outside the p-cell.*

To better understand this essential distinction, in figure 3.3 and 3.4 an example is reported of p-cells inner and outer oriented, respectively. As it can be seen, the cells orientation of figure 3.3 is built by considering only points (for the 1-cell) or edges (for the 2-cell) or surfaces (3-cell) that belong to the considered cell to be oriented. Differently, in figure 3.4 to deduce the orientation we need some external references as for instance a line crossing a 2-cell to be oriented. We remark that it is always possible to deduce the inner orientation of a cell from its outer orientation thanks to the *screw* rule, and vice versa.

With respect to these two figures, we specify, for the sake of clearness, that some *simplices* of $\mathbb{R}^3$ are depicted: a *simplicial* cell is the simplest geometrical cell of a given p dimension that can be conceived in a given space. Since we are working in the 3D space, the simplicial 3-cells are represented by tetrahedrons whose *faces* are simplicial 2-cells. It follows that a simplicial mesh or grids is any partition of the space realized with only triangles (for 2D geometries) or tetrahedra (for 3D geometries).

It is now worth to introduce the following definition:

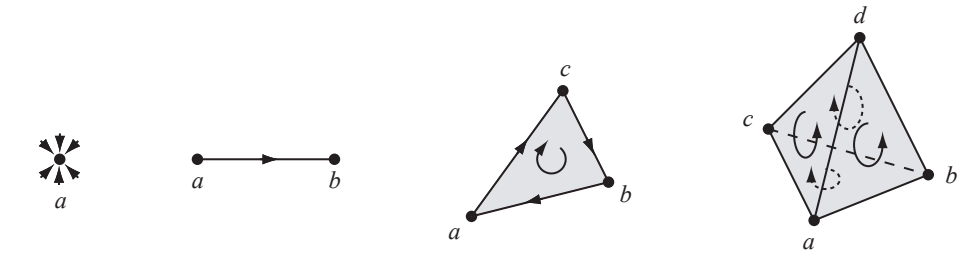

Figure 3.3: Inner orientation of simplices in $\mathbb{R}^3$.

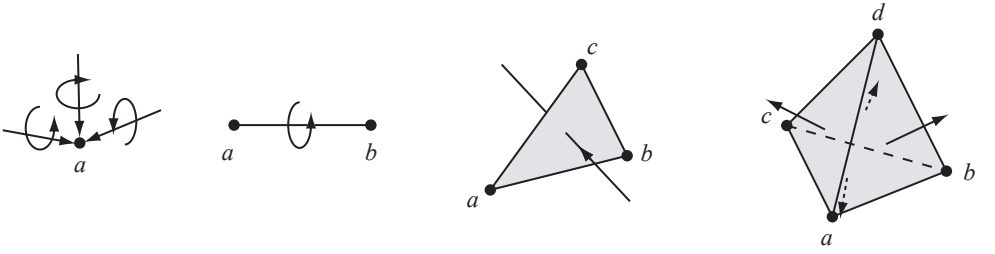

Figure 3.4: Outer orientation of simplices in $\mathbb{R}^3$.

Definition 9. *Given a p-cell, the cell* ***faces*** *are those* $(p-1)$*-cells that are incident to the p-cell.*

Thus, for example, the faces of a simplicial 2-cell are their three bounding edges. The collection of faces of an assigned p-cell is called *boundary* of the p-cell.

On the other hand, for analogy, we can also define the concept of *coface*; in fact,

Definition 10. *In a cell complex* $\mathcal{K}$*, we call* ***cofaces*** *of a p-cell the* $(p+1)$*-cells that share the considered common cell.*

An intuitive representation of this operation is depicted in figure 3.5. As it was done before, the collection of cofaces of a p-cell is denoted as *coboundary*; we remark that a coboundary cannot be defined on a cell complex constituted by cells of a unique p type.

The dual complex $\tilde{\mathcal{K}}$

Once a cell complex $\mathcal{K}$ is defined, it is possible to obtain another cell complex that will be called *dual* and it will be denoted as $\tilde{\mathcal{K}}$ from a partition of the initial *primal* complex $\mathcal{K}$. This partition can be realized in different ways but it has to satisfy the general property such that the p-cells of $\tilde{\mathcal{K}}$ have to be in a *one-to-one* correspondence with the $(n-p)$-cells of $\mathcal{K}$, where n is the space dimension of $\mathcal{K}$. As a consequence:

Property. *The intersection between a p-cell denoted* σ_w^p *of* $\mathcal{K}$ *and a dual* $(n-p)$*-cell denoted* $\tilde{\sigma}_j^{n-p}$ *of* $\tilde{\mathcal{K}}$ *is a point if* $w = j$ *or it is empty if* $w \neq j$*.*

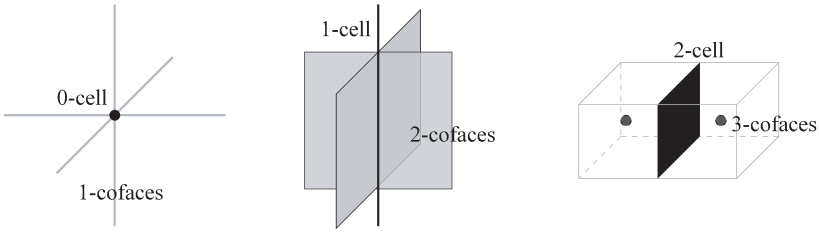

Figure 3.5: Coboundary of p-cells.

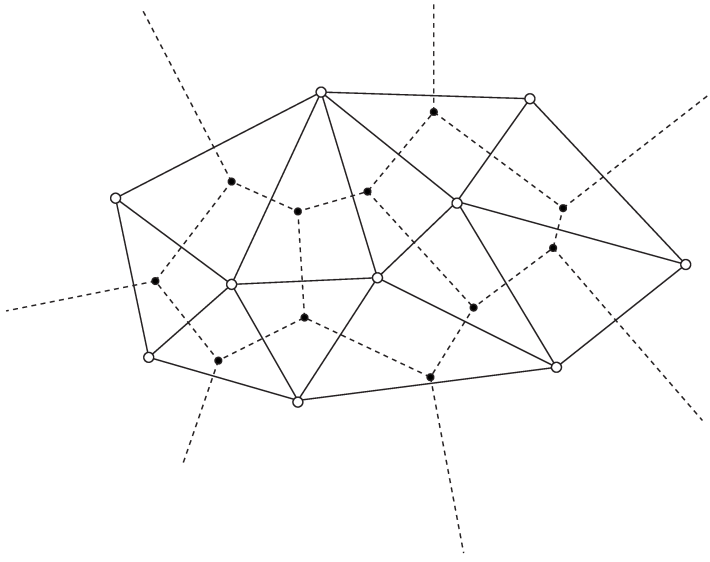

Figure 3.6: Voronoï dual complex on a Delaunay triangulation

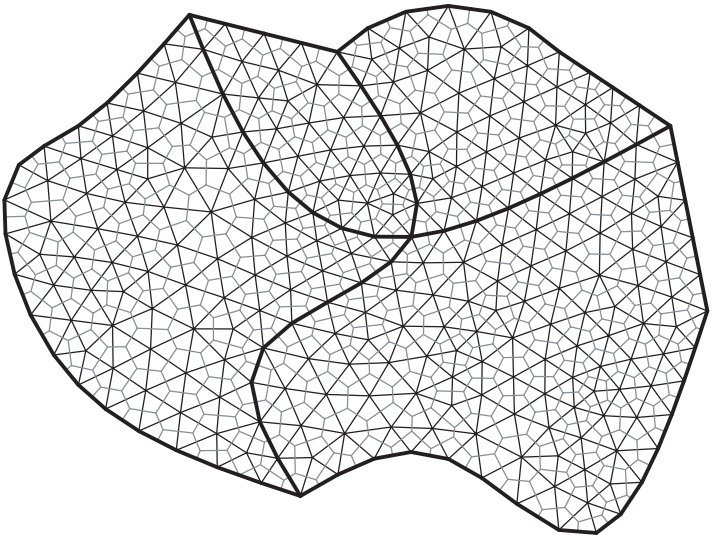

Figure 3.7: Barycentric dual complex $\tilde{\mathcal{K}}$ of a 2D simplicial complex $\mathcal{K}$. The dual of a simplicial complex is not simplicial anymore.

A very popular choice to obtain such a partition is based on **Voronoï diagram** applied to a Delaunay triangulation [15]. An example of this partition for a simplicial 2D complex is shown in figure 3.6, from which we can notice that is based on triangles circumcenters; in fact, since circumcenters have the property of belonging to the axis of the edges of the triangle, once they are connected one to the other they form new straight edges on the dual complex and in addition they subdivide each primal edge into two equal parts. All these pleasing properties make this partition very appealing for CEM since this widely simplify the discrete relations between EM fields when written on a discrete grid. Yet, this dual partition has also some drawbacks: first, when applied to a Delaunay triangulation it could happen that a circumcenter is sited outside the triangle (figure 3.8b) thus leading to have additional hurdles during the computation [16]; second, if the orientation of the dual edge having two circumcenters as endpoints changes (figure 3.8c) then the matrix of the problem does not result to be symmetric and positive definite; third, a Delaunay partition can be performed only on simplicial meshes since circumcenters are defined only on triangles and then extended to tetrahedra. This last limitation in particular cannot be neglected since we will mainly work on polyhedral grids.

For this reason, another criteria that can save most of the useful feature of Voronoï partition is necessary. Historically, the solution was found thanks to the introduction of a *barycentrical* subdivision [17]. In fact, if instead of circumcenters we make use of barycenters we can achieve three important results:

- the barycenter of a p-cell is always contained inside the cell itself;
- a barycentric partition can be extended to polygons and polyhedron;
- despite the fact that on a barycentric dual grid the dual edges are not perpendicular

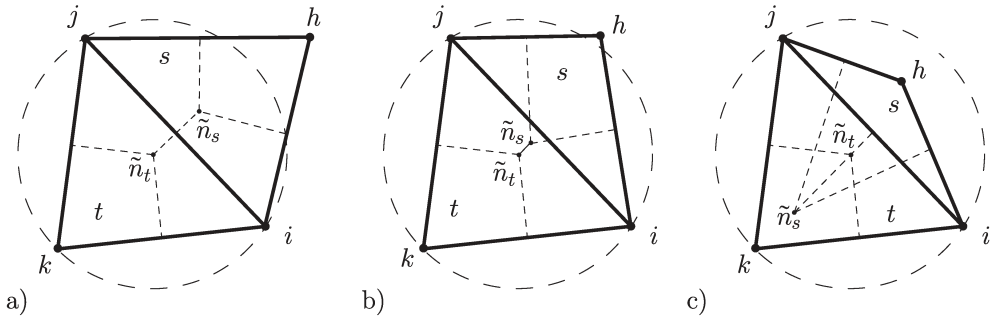

Figure 3.8: Possible circumcenters placing on a triangulation. a) Delaunay triangulation with the circumcenters $\tilde{n}_t, \tilde{n}_s$ inside the surface of t and s triangles, respectively. b) Delaunay triangulation with $\tilde{n}_s$ outside the triangle s. The orientation of the edge $(\tilde{n}_t, \tilde{n}_s)$ does not change. c) Non Delaunay triangulation: the orientation of the edge $(\tilde{n}_t, \tilde{n}_s)$ changes.

to primal edges anymore, they still cross them in a unique point and they subdivide them into two equal parts.

All these features are shown in figure 3.7 for a simplicial complex.

At this point, given an arbitrary primal grid $\mathcal{K} \in \mathbb{R}^3$ we will denote with n its primal nodes, with e its primal edges, with f its primal faces and with v its primal volumes. On the other hand, on the barycentric dual grid $\tilde{\mathcal{K}}$, we will denote with $\tilde{n}$ the dual volumes in a one-to-one correspondence with the n primal nodes, with $\tilde{e}$ the dual faces in a one-to-one correspondence with the e primal edges, with $\tilde{f}$ the dual edges in a one-to-one correspondence with the f primal faces and with $\tilde{v}$ the dual nodes in a one-to-one correspondence with the v primal volumes. The outer orientation of $\tilde{\mathcal{K}}$ is inherited from the inner orientation of the primal grid by means of the screw rule.

3.1.1 Primal and dual time elements

Beyond space elements described as cell complexes, also time instants and time intervals are necessary to obtain a framework on which Maxwell's equation can be reformulated. To this purpose, we here also introduce a *time grid*. This grid will not be directly involved in the solution of our eddy current problem since we will work only in the frequency domain but it is still required if we want to provide a general framework for Maxwell's equations.

For this reason, we will denote as I a time instant and with T a time interval each of them endowed with an inner orientation. If we rely on the usual representation of time on a straight line, as in figure 3.9, we can notice that I and T belongs to two interlocked grids. This inspires the construction of a further grid of time whose elements denoted $\tilde{I}$ and $\tilde{T}$ are in a one-to-one correspondence respectively with T and I thus configuring this second grid as a dual time grid endowed with an outer orientation inferred by the primal grid. By this way, to fulfil the oddness property of equations, we can obtain two groups of time elements: the first group $(I, \tilde{T})$ and its dual $(\tilde{I}, T)$. In the present section,

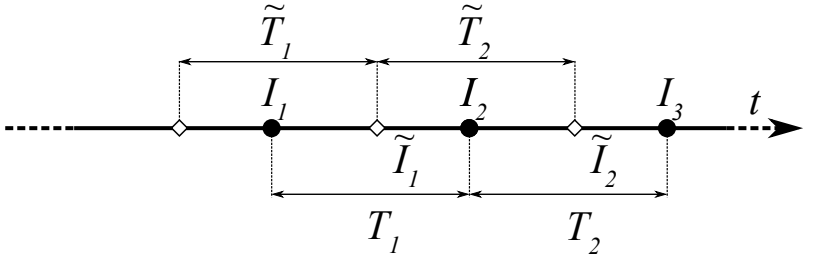

Figure 3.9: Primal instants and intervals I and T and related dual elements $\tilde{I}$ and $\tilde{T}$.

we will not give more details about this classification that will become meaningful in the continuation.

3.2 Algebraic topology

After introducing primal and dual complexes with their inner and outer orientation, we are now ready to describe the algebraic structure with which the aforementioned geometrical entities are treated. This will be done thanks to *algebraic topology* a science that studies the topological property of geometrical entities that are invariant when an homeomorphism is applied; the geometrical entities to which algebraic topology refers are called *topological manifolds*: in short, a manifold is a mathematical space in which every point has a neighborhood that is homeomorphic to an Euclidean space. In other words, this allows to study the properties of complicate geometrical structures by reducing them to more abstract entities. For instance, curves are 1-manifolds, surfaces are 2-manifolds and so on. As a consequence, it becomes natural to associate manifolds to cell complexes and utilize the same algebra to describe them.

3.2.1 Incidence matrices

The first essential algebraic structures to be introduced are undoubtedly incidence matrices that contain information about the topological connections between different kinds of oriented p-cells. The incidence number between a p-cell denoted σ_i^p and another $(p-1)$-cell σ_j^{p-1} to be encoded in the matrix $\mathcal{M}$ can be determined as

$$\mathcal{M}(i,j) = \begin{cases} +1 & \text{if } \sigma_i^p \text{ is a coface of } \sigma_j^{p-1} \text{ and their orientation is compatible} \\ -1 & \text{if } \sigma_i^p \text{ is a coface of } \sigma_j^{p-1} \text{ and their orientation is not compatible} \\ 0 & \text{if } \sigma_i^p \text{ is not a coface of } \sigma_j^{p-1} \end{cases}$$

Given the cell complex $\mathcal{K} \in \mathbb{R}^3$, starting from a global numbering typically assigned to its 0-cells (i.e. the grid nodes) and following the previous definition, it is possible to construct the edges-nodes incidence matrix $\mathbf{G}$. The arbitrary numbering assigned to the edges, will then be used to encode the faces-edges matrix $\mathbf{C}$ which will be the reference to build the volumes-faces matrix $\mathbf{D}$. These three incidence matrices completely describe the topological relations between the oriented p-cells of $\mathcal{K}$.

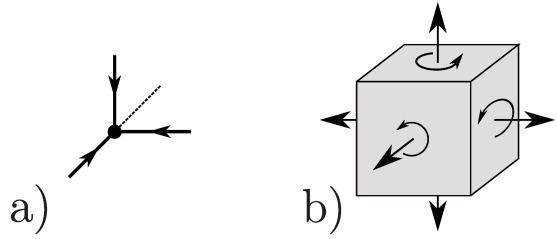

Figure 3.10: a) a cell σ^0 oriented as a sink; b) a cell σ^3 oriented as a source.

To determine whether the orientation between a cell and its boundary elements is compatible, the screw rule has to be applied: if the inner orientation of a p-cell σ_i^p induces an outer orientation on the incident $(p-1)$-cell σ_j^{p-1} which is compatible with its previously assigned inner orientation via the screw rule, then σ_i^p and σ_j^{p-1} will have compatible orientations. Consequently, this procedure requires to assign an arbitrary inner orientation to all the cells of $\mathcal{K}$ before constructing the incidence matrices. To that end, the typical convention is to assign to the nodes n an inner orientation as *sinks* and to the volumes v an inner orientation as *sources*, see figure 3.10; this practice allows to chose for edges and faces an arbitrary orientation that will be then automatically and coherently encoded into $\mathbf{D}$, $\mathbf{C}$ and $\mathbf{G}$.

Incidence matrices of the dual complex The construction of incidence matrices $\tilde{\mathbf{G}}$, $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{D}}$, referred to the dual complex can easily be deduced by the primal ones. In fact, thanks to the one-to-one correspondence between cells of $\mathcal{K}$ and $\tilde{\mathcal{K}}$ and thanks to the construction of the orientation of the dual grid based on the outer orientation inferred by the primal cell complex, we have the following equalities:

- $\tilde{\mathbf{G}} = -\mathbf{D}^T$;
- $\tilde{\mathbf{C}} = \mathbf{C}^T$;
- $\tilde{\mathbf{D}} = \mathbf{G}^T$.

The *minus* sign in the right hand side of the first equation is a direct consequence of the choice of endowing primal nodes of a sink orientation and primal volumes of a source orientation.

Being aware that a complete comprehension of these concepts may require several months of readings and practice, we here suggest to the curious reader that a more exhaustive explanation can be found in [18] or more specifically in [13] pp. 185–190.

3.2.2 Discrete forms

The last step to be done before obtaining a discrete mathematical approximation of the physics of EM fields is the introduction and description of discrete forms. To understand what these forms are, we have to take a detour, move to mathematics and mention exterior differential forms [19, 20]. Historically, in fact, they represent the main contribution

that has changed the way of conceiving and mathematically expressing fields of physics, moving it from differential calculus to something more related to the very geometrical nature of the fields itself. More precisely, thanks to exterior differential forms the choice of a fixed coordinate system it is not necessary anymore since fields are described as *pure* vectors in the space, on which some geometrical differential operations are defined to describe the divergence, curl and gradient as continuous operations on the vectors and not on their components that rely on a coordinate reference [21, 22].

With respect to what has just been said, **discrete** forms add another feature: the possibility of defining algebraic operations that act directly on the *discrete* and *finite* geometrical entities of a given discretization grid, i.e. a cell complex $\mathcal{K}$, thus resulting to be the perfect candidate to deal with CEM.

Chains

In order to approximate EM fields, it is possible to build an abelian group based on cell complexes. In fact, if p-cells are chosen as a basis of this group, we have that

Definition 11. *A p-dimensional chain c_p or **p-chain** is a integer linear combination of p-cells of a cell complex having an inner or outer orientation ($\mathcal{K}$ or $\tilde{\mathcal{K}}$) each multiplied by an integer $d \in \mathbb{Z}$.*

Formally:

$$c_p := \sum_{k=1}^{N} d_k \sigma_k^p \tag{3.1}$$

or also with the array

$$c_p := [d_1, d_2, ..., d_N]^T . \tag{3.2}$$

The integer d_k is called *multiplicity* of the σ_k^p cell. It is then remarked the following property

$$d_k(-\sigma^p) = (-d_k)\sigma_k^p$$

to preserve the oddness of the EM fields to be described[1].

It is also true that given two p-chains belonging to the same cell complex of p-cells, c_p and e_p they can be added as:

$$c_p + e_p = \sum_{k=1}^{N} d_k \sigma_k^p + \sum_{k=1}^{N} n_k \sigma_k^p = \sum_{k=1}^{N} (d_k + n_k)\, \sigma_k^p$$

from which the commutative property $c_p + e_p = e_p + c_p$ directly follows.

As a final remark, it is here recalled that a p-chain of the form $[0, 1, ..., 0]^T$ with a single multiplicity equal to one in correspondence of the i-th entry of the array is called *ith*-elementary chain.

In conclusion, by this way, it has been shown that it is possible to describe portion of the computational domain as a sum of volumes, surfaces, segments or nodes depending on the situation.

[1] This requirement arises by the fact that changing the orientation of a vectorial field implies a change on the direction of its effects.

Boundary of a p-chain We here consider two chains c_p and c_{p-1} defined on two different p-cells types so that one group of cells is the boundary of the other. It is possible to define an operator ∂_p that will be called *boundary operator* as a map acting on c_p so that

$$c_{p-1} = \partial_p c_p. \tag{3.3}$$

This means finding a map that can determine the multiplicities (also called *coefficients*) of c_{p-1} starting from the coefficients of c_p accordingly with their orientation. But this map has already been introduced and it can be identified as the transpose of an incidence matrix $\mathcal{M}$ of $\mathcal{K}$. As a consequence, working in $\Omega \in \mathbb{R}^3$, we have

$$\partial_1 = \mathbf{G}^T \tag{3.4a}$$
$$\partial_2 = \mathbf{C}^T \tag{3.4b}$$
$$\partial_3 = \mathbf{D}^T. \tag{3.4c}$$

It can be shown that the following property is satisfied:

Property. *The boundary of a boundary is a null chain*

$$\partial_{p-1}(\partial_p c_p) = 0_{p-2} \quad \forall c_p$$

that in terms of incidence matrices reads as

$$\mathbf{C}^T\mathbf{D}^T = \mathbf{0} \tag{3.5a}$$
$$\mathbf{G}^T\mathbf{C}^T = \mathbf{0}. \tag{3.5b}$$

These last equations already reflects two fundamental relations of vectorial calculus that will be later recalled.

Cochains

After having introduced chains, it is now possible to provide a definition for discrete forms which algebraic topology refers to as *cochains* due to their duality with chains. In fact,

Definition 12. *Given a cell complex endowed with either an inner or outer orientation and an abelian group $\mathcal{A}$ whose elements can be scalar, vectors or matrices, a* ***discrete p-form*** *or* ***p-cochain****, denoted in the following as $\boldsymbol{c}^p$, is a map that associate p-chains c_p to elements of $\mathcal{A}$.*

This mapping from c_p to $\boldsymbol{c}^p$ will be formally denoted as $\langle \boldsymbol{c}^p, c_p \rangle$ which reads as *the value of the cochain $\boldsymbol{c}^p$ evaluated on c_p*. With this formalism it is then possible to state the following property.

Property. *A discrete p-form is an additive and homogeneous map.*

Symbolically:

$$\langle \boldsymbol{c}^p, c_p + {c'}_p \rangle = \langle \boldsymbol{c}^p, c_p \rangle + \langle \boldsymbol{c}^p, {c'}_p \rangle$$
$$\langle \boldsymbol{c}^p, k c_p \rangle = k \langle \boldsymbol{c}^p, c_p \rangle.$$

Similarly to chains, the values of a p-form $\boldsymbol{c}^p$ computed on a p-chain c_p can also be represented as a vector as

$$\boldsymbol{c}^p = [v_1, v_2, \cdots, v_N]$$

with N the number of p-cells of c_p.

As a consequence, the value v of $\boldsymbol{c}^p$ can be defined as the sum of the values v_i on each p-cell C_j^p of c_p multiplied for the chain multiplicity d_j. This leads to

$$\langle \boldsymbol{c}^p, c_p \rangle = \sum_{k=1}^{N} v_k d_k = [v_1, v_2, \cdots, v_N] \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix}. \tag{3.6}$$

Coboundary of a p-cochain In a similar way to what has been done for chains, it is possible to define an operator also for differential forms that will then be called *coboundary* of $\boldsymbol{c}^p$ and denoted as δ^p. Its effect on $\boldsymbol{c}^p$ can be described as the following list of actions:

- given a p-form on an oriented p-chain, select the $(p+1)$-coboundary of the chain;
- transfer the values v_i of the cochain to the chain coboundary cells considering their orientation by means of incidence matrices;
- sum the transferred values on each $(p+1)$-cells thus obtaining the value w of a new $(p+1)$-form, $\boldsymbol{c}^{p+1}$.

This process can be formalized in the next definition.

Definition 13. *Given a p-form $\boldsymbol{c}^p$ and a $(p+1)$-form $\boldsymbol{c}^{p+1}$ and denoting with v and w their values $v = \langle \boldsymbol{c}^p, c_p \rangle$, $w = \langle \boldsymbol{c}^{p+1}, c_{p+1} \rangle$ computed on the two chains c_p and c_{p+1}, the value w of $\boldsymbol{c}^{p+1}$ can be obtained, as previously described, from $\boldsymbol{c}^p$ thanks to the coboundary operator:*

$$\boldsymbol{c}^{p+1} = \delta^p \boldsymbol{c}^p.$$

As for the boundary operator ∂_p, also the just described coboundary process can be expressed by means of an incidence matrix applied to a discrete form in its vectorial notation. Then, we will have the following correspondences:

- $\delta^0 = \mathbf{G}$;
- $\delta^1 = \mathbf{C}$;
- $\delta^2 = \mathbf{D}$.

Moreover, if we keep reasoning with the dualism with respect to p-chains, it is not surprising that:

Property. *The coboundary of the coboundary of $\boldsymbol{c}^p$ is a null discrete form*

$$\delta^{p+1}\left(\delta^p \boldsymbol{c}^p\right) = \mathbf{0}^{p+2} \quad \forall \boldsymbol{c}^p \tag{3.7}$$

This last property can also be expressed by using incidence matrices as

$$\mathbf{DC} = \mathbf{0}, \qquad \mathbf{CG} = \mathbf{0}. \tag{3.8}$$

Discrete Stokes' theorem The fundamental theorems of vectorial calculus can be reformulated by using algebraic topology thanks to the coboundary operator in a unique theorem called Generalized Stokes' Theorem. This one is the discrete version of its analogous expressed by using differential p-forms on manifolds that can be found in [22] pp. 553–560. In the discrete frameworks of chains and cochains, it reads as

Theorem. *The value of a discrete p-form $\boldsymbol{c}^p$ on the $(p+1)$-boundary of a $(p+1)$-chain c_{p+1} is equal to the value of the $(p+1)$-dimension coboundary of $\boldsymbol{c}^p$ evaluated on a correspondent $(p+1)$-chain.*

This also writes as:

$$\langle \boldsymbol{c}^p, \partial_{p+1} c_{p+1} \rangle = \langle \delta^p \boldsymbol{c}^p, c_{p+1} \rangle. \tag{3.9}$$

The theorem, whose proof is not reported for the sake of brevity, returns the Fundamental Theorem of Calculus for $p = 0$, for $p = 1$ it gives the classical Stokes' Theorem, for $p = 2$ Gauss' Divergence Theorem is expressed instead.

The main consequence is that the Generalized Stokes' Theorem strengthens the idea of considering incidence matrices as discrete differential operators thus revealing the choice of the formalism previously used for them. In fact, the incidence matrix $\mathbf{G}$ can be considered as the discrete version of the gradient *grad* or $\nabla\bullet$ differential operator, $\mathbf{C}$ is the discrete version of the differential curl *curl* or $\nabla \times (\bullet)$ operator whereas $\mathbf{D}$ stands for the discrete version of the differential divergence *div* or $\nabla \cdot (\bullet)$. It follows that the two relations of (3.8) which are equivalent to (3.5) are the discrete version of the vectorial calculus identities $div(curl(\bullet)) = 0$ and $curl(grad(\bullet)) = 0$.

3.3 Tonti's diagram

The final aim of this chapter was twofold: on one hand, an alternative way of describing physics by relying on a discrete framework based on p-chains and p-forms has already been delineated, on the other hand we still have to provide a proper and definitive recasting of Maxwell's equations. To this goal, we will here try to write them on this discrete structure by starting with a classification of the physical variables to then explain how EM laws can be represented into a diagram, called *Tonti's diagram*, as proposed in [13] pp. 221–223. Once the rationale of the diagram structure will be clear, it will be populated and adapted to describe an eddy current integral formulation of which the main topics of this thesis debates.

3.3.1 Physical variables

Vectorial calculus does not reflect the real nature of physics. In fact, due to its formalism, it requires the definition of vectorial fields that exists in each point of the space for a given instant of time or the definition of quantity densities belonging to an (ideal) infinitesimal portion of the space.

This depiction is extremely far from the actual experience with electromagnetic phenomena that, in the real world, can be verified just by means of *measurement procedures*. For example, to detect the presence of a magnetic induction field $\boldsymbol{b}(\boldsymbol{r}, t)$ a common approach could be represented by the usage of a magnetic induction field probe which does not directly "feel" the presence of $\boldsymbol{b}(\boldsymbol{r}, t)$ but it presumes its existence by measuring the effect of an induced voltage on a coil by exploiting Faraday–Neumann law. In order words, we know about the existence of a magnetic induction field by measuring the *flux* of that field through a surface (of the coil) and by reading the value of a voltage: for the former term we are thus considering the *surface integral* of $\boldsymbol{b}$ whereas for the latter we are looking at the *line integral* of $\boldsymbol{e}(\boldsymbol{r}, t)$ along the coil of the magnetic induction probe. The whole scenario tell us that usually, for practical purposes, we commonly deal with *global quantities* instead of their respective field variables.

Since CEM is undoubtedly closer to a measurement procedure than to a conceptual mathematical modelling of physics, it seems natural to work with global quantities, also referred to as integral variables, for our numerical model too. Thus, given that the computational domain has been discretized to be represented thanks to algebraic topology with chains and cochains, we can exploit the definition of discrete p-forms to associate a specific EM global quantity to a precise p-chain of the primal or dual cell complex $\mathcal{K}$ and $\tilde{\mathcal{K}}$. At this point, a distinction between different categories of global quantities is necessary in order to discern on which cell complex each variable can be associated. Physical variables can be grouped into:

- *configuration* variables;
- *source* variables;
- *energetic* variables.

The first kind is related to problem equations that describe the shape of the field and the mutual relationship between different fields. In dealing with eddy currents, whose equations have already been listed in section 2.2.1, these variables are the magnetic induction $\boldsymbol{b}$ and the electric field $\boldsymbol{e}$. On the other hand, we have source variables that in our case are identifiable with the magnetic field $\boldsymbol{h}$ and with the current density $\boldsymbol{j}$: these variables explain the causality of EM phenomena. Finally, energetic variables are a product of the former two and thus they clearly deal with the definition of field energy that should be familiar to the reader (for instance, the magnetic field energy density is proportional to the product between $\boldsymbol{b}$ and $\boldsymbol{h}$).

The proposed classification of the variable it is not an end in itself since cell method practice has shown that there is a strict relation between the type of a variable and its position on the final diagram we are meant to obtain. Generally, configuration and source variables are associated to distinct complexes in a way that it is possible to define a *consistent* map from one variable of a complex to another of its dual.

3.3.2 Physical laws

A schematic of the diagram we want to achieve is represented in figure 3.11. From its detailed description it is possible to allocate, identify and organize all the missing elements we need.

The main structure of the diagram is organized as follows:

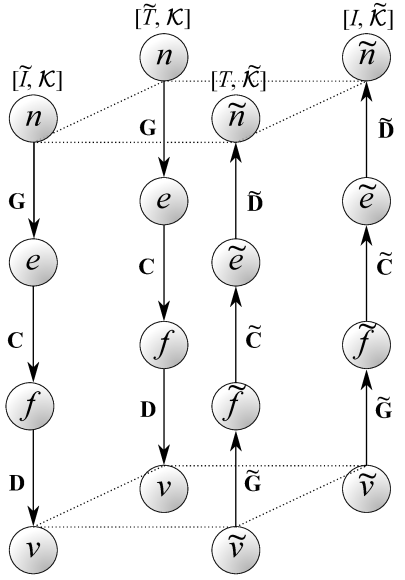

Figure 3.11: Basic structure of Tonti's diagram. All cells types of the primal and dual complex are showcased and related to either a time instant or interval.

- it has a 3D shape to take into consideration both time and space complexes;
- on the left, the cells of the primal complex $\mathcal{K}$ can be found, on the right we find their duals on the dual complex $\tilde{\mathcal{K}}$;
- similarly, on the right the cells of the primal time complex I and T i.e. primal instants and intervals are placed and they are thus associated to $\tilde{\mathcal{K}}$ whereas on the left the same frame is replicated for dual time instants and intervals ($\tilde{I}$ and $\tilde{T}$) associated to $\mathcal{K}$;
- the vertical hierarchy of each of the four columns considers the cells dimension starting from the lower one constituted by 0-cells i.e. the nodes n up to 3-cells alias volumes v;
- along the horizontal direction we can appreciate the one-to-one correspondence between primal and dual elements, for instance on the left frontal columns we can found primal nodes and dual time instants $[n, \tilde{I}]$ whereas at the same level on the right we have their dual i.e. dual volumes and primal time intervals $[\tilde{n}, T]$[2].

The successive consideration to be done regards movements inside the diagram. In fact, the main purpose of the diagram is describing a physical law by moving along the diagram starting from one of its nodes and then changing the position by following a vertical or horizontal direction. Specifically, vertical movements happen inside the same

[2]For the primal-dual correspondence explanation between the time grids we here recall figure 3.9 for the sake of precision.

cell complex while in contrast horizontal ones concern the change of the complex from primal to dual and vice versa. Moreover, if global physical quantities are associated to a p-space complex thanks to discrete p-forms it is thus clear that vertical movements deal with the coboundary operator related to incidence matrices as explained in the previous sections of this chapter and they are allowed just from a p to a $(p+1)$ cell-form. In conclusion, we can for instance move from a global quantity associated to primal nodes to another quantity associated to primal edges thanks to $\mathbf{G}$ incidence matrix: in other words, we change the position on the diagram by applying a discrete *gradient* operator to the starting global quantity.

As far as horizontal displacements are concerned we have two possibilities: movements between frontal and backward planes deal with the *time derivative* operator whereas that ones between two columns on the same plane are linked to the so called *Hodge operator* $\star$ [23] of which we will soon debate.

The identification of time derivatives with movements between the two planes directly comes from the very definition of instants and intervals on the time complex and from the definition of finite differences. In fact, since a finite difference in time domain concerns considering a function at two distinct instant to then sum the contributions taking into account both the function and the time orientations, this can be realized on the time complex via a coboundary operation between time instants and time intervals thus explaining the rationale applied to the diagram.

Furthermore, we remark again the duality between discrete p-forms here considered for the proposed diagram and the differential p-forms: indeed, the same diagram can be obtained with a differential formalism as it was done by A. Bossavit with the *Maxwell's house* in his seminal work *"On the geometry of electromagnetism"* [24].

3.3.3 Constitutive laws

Going back to chapter 2, we newly focus our attention on constitutive relations of equation (2.2) and (2.3). Indeed, if for the sake of brevity we just focus on the necessary constitutive relations of an eddy current problem, it will be noticed that the two fields involved in each equations and linked by the material parameter belong to different classification classes. Specifically, in the magnetic constitutive law

$$\boldsymbol{b}(\boldsymbol{r},t) = \mu(\boldsymbol{r})\boldsymbol{h}(\boldsymbol{r},t) \tag{3.10}$$

the magnetic induction field $\boldsymbol{b}$ belongs to the class of configuration variables whereas the magnetic field $\boldsymbol{h}$ comes from the class of source variables. Similarly, in the conduction constitutive law

$$\boldsymbol{e}(\boldsymbol{r},t) = \rho(\boldsymbol{r})\boldsymbol{j}_c(\boldsymbol{r},t) \tag{3.11}$$

we have again that the configuration variable $\boldsymbol{e}$ is linked to the source variable $\boldsymbol{j}_c$.

As a consequence, continuing the reasoning of the preceding section and since it has been said that configuration and source variables belongs to the two different complexes $\mathcal{K}$ and $\tilde{\mathcal{K}}$ of the discretized computational domain, it is natural to identify a strict bound between constitutive relations and the aforementioned Hodge operator [25]. Indeed, its role will be that one of expressing constitutive relations at a discrete geometrical level.

Without going into the details in this paragraph, we anticipate that choices in the

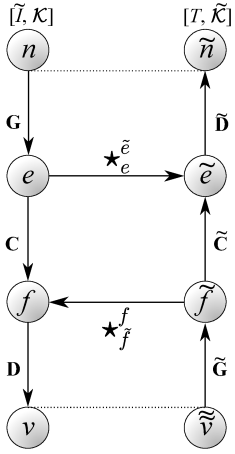

Figure 3.12: Basic structure of Tonti's diagram. The discrete Hodge star operator $\star$ acts on the horizontal direction mapping the two complexes one into the other.

construction of the Hodge operator, also called Hodge *star* operator, differentiate computation methods one from the other. In fact, since constitutive relations involve the metrics of the problem, it will be shown that in order to obtain a discrete $\star$ operator it is necessary to interpolate global variables over the discrete grid by means of shape function, and this can be done in several different ways. For instance, besides the initial formulations based on nodal shape functions [26] the most common approach in the CEM community is for sure represented by the exploitation of Whitney's forms [27, 28] that generate vector functions which are defined over the mesh geometrical elements as nodes, edges and faces; a construction that clearly well fits the algebraic framework so far proposed.

In the next chapters we will treat this topic in a very detailed way, moving from Whitney's forms, that despite their geometrical interpretation [29] have an inner mathematical soul, to a pure geometrical approach. To conclude, in figure 3.12 we propose the frontal plane of Tonti's diagram with the addition of the horizontal mapping between primal and dual grids. As it is shown, the star operator allows the mapping from primal to dual and vice versa thus resulting into a bijective map; unfortunately, this is rarely possible in practice since shape functions are usually defined for primal geometrical elements whereas their dual is not determined because of the barycentric construction of the dual grid causing a splitting of the dual edges that are not segments anymore. This behaviour leads to a discrete Hodge $\star$ preferably directed from the primal to the dual complex as it will happen for the volume integral formulation exposed in the continuation of this work and, as previously suggested, this also influences the positioning of physical variables that has to be in such a way to favour mappings from $\mathcal{K}$ to $\tilde{\mathcal{K}}$ and not vice versa.

3.4 Classification diagram for eddy current integral formulation

Since all the necessary tools have been exposed, it has come the moment of detailing Tonti's diagram for an eddy current problem expressed by means of an integral formulation using the geometrical framework based on discrete p-forms on a primal and dual cell complex. The goal will be identifying the required global physical quantities involved and establish in which complex they act to then represent the equations of magnetodynamics on the diagram.

During this phase, we will still not provide all the details underneath the construction of the discrete Hodge operator since our target is just the equations representation on the geometrical framework so far introduced and not the solution of the problem. Indeed, next chapter will be devoted to this purpose.

For the sake of simplicity, the equations we are interested in are again reported in the following:

$$\nabla \times \boldsymbol{h}(\boldsymbol{r},t) = \boldsymbol{j}_c(\boldsymbol{r},t), \tag{3.12}$$
$$\nabla \times \boldsymbol{e}(\boldsymbol{r},t) + \partial_t \boldsymbol{b}(\boldsymbol{r},t) = \boldsymbol{0}, \tag{3.13}$$
$$\nabla \cdot \boldsymbol{b}(\boldsymbol{r},t) = 0, \tag{3.14}$$
$$\boldsymbol{b}(\boldsymbol{r},t) = \mu(\boldsymbol{r})\boldsymbol{h}(\boldsymbol{r},t) \tag{3.15}$$
$$\boldsymbol{e}(\boldsymbol{r},t) = \rho(\boldsymbol{r})\boldsymbol{j}_c(\boldsymbol{r},t) \tag{3.16}$$

in addition with the Biot–Savart law in a uniform non magnetic media ($\mu(\boldsymbol{r}) = \mu_0 \ \forall \boldsymbol{r} \in \Omega$) that relate $\boldsymbol{j}(\boldsymbol{r},t)$ to the magnetic vector potential $\boldsymbol{a}(\boldsymbol{r},t)$ by means of the integral relation:

$$\boldsymbol{a}(\boldsymbol{r},t) = \frac{\mu_0}{4\pi}\int_V \frac{\boldsymbol{j}(\boldsymbol{r}',t)}{|\boldsymbol{r}'-\boldsymbol{r}|}dv.$$

where the magnetic vector potential is a vector field so that $\boldsymbol{b}(\boldsymbol{r},t) = \nabla \times \boldsymbol{a}(\boldsymbol{r},t)$ since $\boldsymbol{b}$ is solenoidal. Similarly, since $\nabla \cdot \boldsymbol{j}_c = 0$ it is possible to define an electric vector potential $\boldsymbol{t}$ so that $\nabla \times \boldsymbol{t}(\boldsymbol{r},t) = \boldsymbol{j}_c(\boldsymbol{r},t)$.

As a consequence of the introduction of the magnetic vector potential another bound between $\boldsymbol{e}$ and $\boldsymbol{a}$ arises: by substituting the definition of $\boldsymbol{a}(\boldsymbol{r},t)$ into (3.13) we have

$$\nabla \times [\boldsymbol{e}(\boldsymbol{r},t) + \partial_t \boldsymbol{a}(\boldsymbol{r},t)] = \boldsymbol{0}$$

which leads to the definition of a scalar electric potential $v(\boldsymbol{r},t)$ so that

$$\boldsymbol{e}(\boldsymbol{r},t) + \partial_t \boldsymbol{a}(\boldsymbol{r},t) = -\nabla v(\boldsymbol{r},t)$$

3.4.1 A discrete de Rham's map

The remaining step is obtaining physical global quantities from the vector fields $\boldsymbol{b}, \boldsymbol{a}, \boldsymbol{e}, \boldsymbol{h}$ and $\boldsymbol{j}_c$ and from the scalar potential v to represent them as discrete p-forms. This goal can be achieved by integrating the vector field on the corresponding p-cells on the primal or dual cell complex.

We notice that this approach can be conceived as an extension to the computational

discrete domain of the *de Rham map* [30] $\mathcal{R}$ linking a differential form ω on a manifold $\mathcal{W}$ to its integral:

$$\begin{aligned} \mathcal{R} : \bigwedge^{p}(\mathcal{W}) &\longrightarrow C^{p}(\mathcal{K}) \\ \omega &\longmapsto \int_{c_p} \omega \end{aligned}$$

Consequently, denoting with C_p the group of p-chains, we will have:

- a discrete 2-form $\mathbf{I}$ representing the integral of the current density $\boldsymbol{j}_c$ on the faces of $\mathcal{K}$, i.e. the current flowing through the grid faces, mapping 2-chains onto real numbers

$$\begin{aligned} \mathbf{I} : C_2 &\longrightarrow \mathbb{R} \\ c_2 &\longmapsto \int_f \boldsymbol{j}_c(\boldsymbol{r}, t) \cdot \hat{\boldsymbol{n}}\, ds; \end{aligned}$$

- a discrete 1-form $\mathbf{T}$ representing the integral of the electric vector potential $\boldsymbol{t}$ along the edges of $\mathcal{K}$, mapping 1-chains onto real numbers

$$\begin{aligned} \mathbf{T} : C_1 &\longrightarrow \mathbb{R} \\ c_1 &\longmapsto \int_e \boldsymbol{t}(\boldsymbol{r}, t) \cdot \hat{\boldsymbol{l}}\, dl; \end{aligned}$$

- a discrete 1-form $\mathbf{F}$ representing the integral of the magnetic field $\boldsymbol{h}$ along the edges of $\mathcal{K}$, alias the magneto motive force *m.m.f*, mapping 1-chains onto real numbers

$$\begin{aligned} \mathbf{F} : C_1 &\longrightarrow \mathbb{R} \\ c_1 &\longmapsto \int_e \boldsymbol{h}(\boldsymbol{r}, t) \cdot \hat{\boldsymbol{l}}\, dl; \end{aligned}$$

- a discrete 2-form $\tilde{\mathbf{\Phi}}$ representing the integral of the magnetic induction field $\boldsymbol{b}$ through the faces of $\tilde{\mathcal{K}}$, i.e. the magnetic flux, mapping 2-chains onto real numbers

$$\begin{aligned} \tilde{\mathbf{\Phi}} : \tilde{C}_2 &\longrightarrow \mathbb{R} \\ c_2 &\longmapsto \int_{\tilde{e}} \boldsymbol{b}(\boldsymbol{r}, t) \cdot \hat{\boldsymbol{n}}\, ds; \end{aligned}$$

- a discrete 1-form $\tilde{\mathbf{U}}$ representing the integral of the electric field $\boldsymbol{e}$ along the edges of $\tilde{\mathcal{K}}$, i.e. the electro motive force *e.m.f.*, mapping 1-chains onto real numbers

$$\begin{aligned} \tilde{\mathbf{U}} : \tilde{C}_1 &\longrightarrow \mathbb{R} \\ c_1 &\longmapsto \int_{\tilde{f}} \boldsymbol{e}(\boldsymbol{r}, t) \cdot \hat{\boldsymbol{l}}\, dl; \end{aligned}$$

- a discrete 1-form $\tilde{\mathbf{A}}$ representing the integral of the magnetic vector potential $\boldsymbol{a}$ along the edges of $\tilde{\mathcal{K}}$, mapping 1-chains onto real numbers

$$\begin{aligned}\tilde{\mathbf{A}} : \tilde{C}_1 &\longrightarrow \mathbb{R} \\ c_1 &\longmapsto \int_{\tilde{f}} \boldsymbol{a}(\boldsymbol{r}, t) \cdot \hat{\boldsymbol{l}}\, dl;\end{aligned}$$

- a discrete 0-form $\tilde{\mathbf{V}}$ representing the scalar potential v on the nodes of $\tilde{\mathcal{K}}$, mapping 0-chains onto real numbers

$$\begin{aligned}\tilde{\mathbf{V}} : \tilde{C}_0 &\longrightarrow \mathbb{R} \\ c_0 &\longmapsto v(\boldsymbol{r}, t).\end{aligned}$$

By doing this we are able to represent EM fields as coefficients of a cell thus obtaining the **degrees of freedom** (DOFs) of the eddy current problem.

In addition to Maxwell's equation we also need the discrete constitutive relation

$$\tilde{\mathbf{U}} = \left[\rho^{\star}_{f,\tilde{f}}\right] \mathbf{I} \tag{3.17}$$

and the integral relation given by Biot–Savart law

$$\tilde{\mathbf{A}} = \left[\mu^{\star}_{f,\tilde{f}}\right] \mathbf{I} \tag{3.18}$$

where $\left[\rho^{\star}_{f,\tilde{f}}\right]$ and $\left[\mu^{\star}_{f,\tilde{f}}\right]$ are matrices representing the discrete version of the Hodge star operator whose computation has still to be detailed.

In conclusion, we come up with the subsequent discrete equations:

- the definition of ungauged magnetic vector potential in the cell complex framework

$$\tilde{\mathbf{C}}\tilde{\mathbf{A}} = \tilde{\boldsymbol{\Phi}} \tag{3.19}$$

- the discrete version of Gauss' law

$$\tilde{\mathbf{D}}\tilde{\boldsymbol{\Phi}} = \mathbf{0} \tag{3.20}$$

- the discrete version of current continuity law

$$\mathbf{D}\mathbf{I} = \mathbf{0} \tag{3.21}$$

- the discrete version of Ampere's law

$$\mathbf{C}\mathbf{T} = \mathbf{I} \tag{3.22}$$

- the discrete relation between $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{A}}$

$$\tilde{\mathbf{U}} + \tilde{\mathbf{G}}\tilde{\mathbf{V}} = -d_t \tilde{\mathbf{A}} \tag{3.23}$$

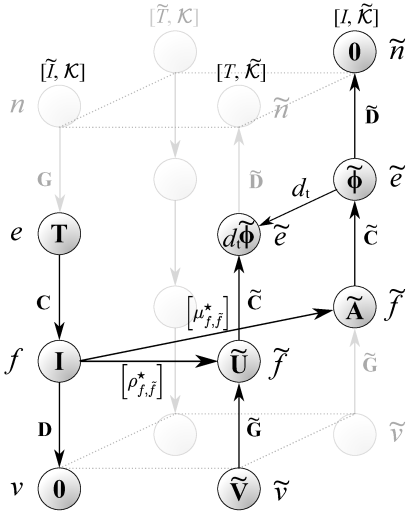

Figure 3.13: Tonti's diagram of the volume integral formulation for eddy currents.

- the discrete version of Faraday–Neumann law

$$\tilde{\mathbf{C}}\tilde{\mathbf{U}} + d_t\tilde{\mathbf{\Phi}} = \mathbf{0}. \tag{3.24}$$

As final remark, we here comment the choice on the DOFs positioning. In fact, once the current $\mathbf{I}$ is placed on the primal complex for the sake of simplicity, it follows that $\mathbf{T}$ and $\mathbf{F}$ must be placed on the same complex since they all are *source* physical variables. On the contrary, $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{\Phi}}$ and, as a consequence, also $\tilde{\mathbf{A}}$ will be assigned to the dual complex $\tilde{\mathcal{K}}$ since they belong to the *configuration* variables class. By doing this, the necessary discrete Hodge operators $\left[\rho^{\star}_{f,\tilde{f}}\right]$ and $\left[\mu^{\star}_{f,\tilde{f}}\right]$ will map a form of the primal grid onto its dual on $\tilde{\mathcal{K}}$, thus respecting the desiderata expressed in the previous section.

The diagram

All the aforementioned equations expressing an eddy current problem with an integral formulation by means of cell method can be represented as a Tonti's diagram as shown in figure 3.13. $\mathbf{T}$ and $\tilde{\mathbf{V}}$ are the input DOFs since they are not dependent by any other global quantity; from them, by moving along the graph arrows we can reconstruct Faraday–Neumann law which will lead to the system of equations to be solved. The far ends of the diagram (in v and $\tilde{n}$) express the current continuity law and Gauss' law mapping $\mathbf{T}$ and $\tilde{\mathbf{\Phi}}$ respectively onto zero by verifying the vectorial identities in their discrete form: $\mathbf{DCT} = \mathbf{0}$ and $\tilde{\mathbf{D}}\tilde{\mathbf{C}}\tilde{\mathbf{A}} = \mathbf{0}$ since $\mathbf{DC} = \mathbf{0}$. In a similar way, we also notice that the contribution of $\tilde{\mathbf{V}}$ in the Faraday-Neumann equation cancels out since $\tilde{\mathbf{C}}\tilde{\mathbf{G}}\tilde{\mathbf{V}} = \mathbf{0}$ thus, from the implementation point of view, the DOFs of the eddy current problem will be represented only by $\mathbf{T}$.

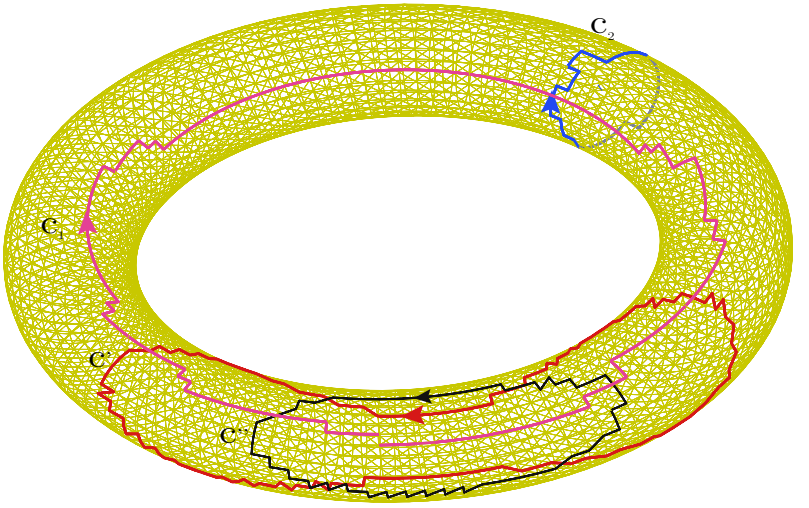

Figure 3.14: Boundary groups and cycles for 1-chains of a torus.

3.5 What is missing: (co)homology theory

There is a slight lie in all the theory so far exposed, specifically about the structure of chains and cochains on a cell complex $\mathcal{K}$. In fact, it has been shown how by defining the incidence matrices $\mathbf{D}$, $\mathbf{C}$ and $\mathbf{G}$ it is possible to express Maxwell's equation on a computational domain Ω without any further specification. Indeed, this sentence it is not correct for all the possible domains as the reader may already know.

Let us consider the torus in figure 3.14 on which it is possible to distinguish between two kinds of paths: the first set containing $\{\boldsymbol{c}', \boldsymbol{c}''\}$ and the second constituted by $\{\boldsymbol{c}_1$ $\boldsymbol{c}_2\}$. It is possible to grasp the characteristic aspect of each of the two group, in fact, the former is constituted by paths that can be expressed by the incidence matrix $\mathbf{C}$ as a linear combination of local paths whereas the latter contains paths that deal with the topological shape of the domain and cannot be expressed by simply using $\mathbf{C}$. This last group is the one we are interested in. From vectorial calculus it is already known that this happens when the computational domain is not *simply connected*, a property that has to be verified for the dissertation of the previous part of this chapter to be valid.

Consequently, our aim is finding a theory that can recognize, categorize and describe such not simply connected geometries by means of algebraic topology. This paramount tool can be identified with homology and co-homology theory [31, 32].

Absolute homology groups

Homology theory is based on the classification of p-chains through the boundary operator ∂. In fact, the following classes exist:

- the class of p-chains c_p that are boundary of a $(p+1)$-chain belongs to the ***bound-***

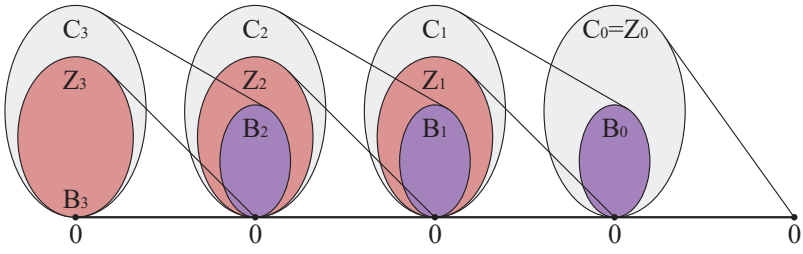

Figure 3.15: Boundary operator over p-chains on the cell complex $\mathcal{K}$.

ary group $B_p(\mathcal{K})$

$$B_p(\mathcal{K}) := \mathrm{cod}(\partial_{p+1}) = \{\partial_{p+1} c_{p+1} \,|\, c_{p+1} \in C_{p+1}(\mathcal{K})\}$$

- the class of p-chains c whose boundary is null and that are not necessarily the boundary of any $(p+1)$-chain belongs to the ***cycle group*** $Z_p(\mathcal{K})$

$$Z_p(\mathcal{K}) := \ker(\partial_p) = \{c_p \in C_p(\mathcal{K}) \,|\, \partial_p c_p = 0\}.$$

which implies that $B_p(\mathcal{K}) \subset Z_p(\mathcal{K})$. The symbol $C_p(\mathcal{K})$ denotes the group of p-chains of the cell complex $\mathcal{K}$.

To these definitions we add the following property:

Property. *Two chains that belongs to the boundary group $B_p(\mathcal{K})$ of the same cell complex $\mathcal{K}$ are* ***homologous****.*

As a consequence of this relation between chains, it is possible to provide the following definition:

Definition 14. *The* ***absolute homology group*** *$H_p(\mathcal{K})$ on the cell complex $\mathcal{K}$ is the quotient*

$$H_p(\mathcal{K}) := Z_p(\mathcal{K})/B_p(\mathcal{K}). \tag{3.25}$$

This group makes a distinction between that p-chains that are boundaries of any $(p+1)$-chains and that other p-chains that are not. This relation is well represented in figure 3.15 where starting from 3-chains the boundary operator ∂_p is applied highlighting all the gaps between Z_p and B_p: that gaps belong to the homology groups.

Relative homology In some cases it is also necessary to limit the topological classification of the computational domain Ω to a subsets of cells of $\mathcal{K}$. This results in *relative* homology which is hereafter described.

Definition 15. *The group of relative p-chains modulo L of the cell complex $\mathcal{K}$ are*

$$C_p(\mathcal{K}, L) := C_p(\mathcal{K})/C_p(L).$$

Consequently, it is possible to define the relative homology group modulo L, $H_p(\mathcal{K}, L)$ as

$$H_p(\mathcal{K}, L) := Z_p(\mathcal{K}, L)/B_p(\mathcal{K}, L). \tag{3.26}$$

As final comment, thanks to these definitions we can now state that, with respect to figure 3.14, cycles $\{c_1, c_2\}$ are generators of the homology group $H_1(\mathcal{K})$ whereas $\{c', c''\}$ are two homologous cycles that belongs to $B_p(\mathcal{K})$.

An extension to discrete forms: cohomology groups

We now want to extend the chains classification just exposed to discrete p-forms since the discrete approximation of an EM problem relies on them. To that end, we here recall the information in section 3.2.2 regarding the duality between the boundary and coboundary operators; in particular, it is true that

$$\partial_{p+1}^T = \delta^p$$

from which it followed the use of incidence matrices as discrete coboundary operator.

Starting from this equality, we can proceed as done for the boundary operator applied to chains and extend the already defined homology classes to the coboundary operator. In fact, we can define:

- $B^p(\mathcal{K}) = \mathrm{cod}(\delta^{p-1}) = \mathrm{cod}(\partial_p)$ as the group of ***p*-coboundaries**;
- $Z^p(\mathcal{K}) = \ker(\delta^p) = \ker(\partial_{p+1})$ as the group of ***p*-cocycles**.

Naturally, it then comes the cohomology group definition.

Definition 16. *The **absolute cohomology group** $H^p(\mathcal{K})$ on the cell complex $\mathcal{K}$ is the quotient group*

$$H^p(\mathcal{K}) := Z^p(\mathcal{K})/B^p(\mathcal{K}). \tag{3.27}$$

As it has been done for the homology group, a relative cohomology group modulo L $H^p(\mathcal{K}, L)$ can be determined too.

The present classification applied to discrete forms leads to the chart in figure 3.16.

Poincaré–Lefschetz duality In order to apply (co)homology to both primal and dual complex it is worth the following theorem.

Theorem. *Given two absolute homology and cohomology groups on $\mathcal{K} \in \mathbb{R}^3$ and their correspondent relative groups on the boundary $\partial\mathcal{K}$ of $\mathcal{K}$, the following isomorphisms are true:*

$$H^p(\mathcal{K}, \partial\mathcal{K}) \approx H_{3-p}(\mathcal{K})$$
$$H_p(\mathcal{K}, \partial\mathcal{K}) \approx H^{3-p}(\mathcal{K}).$$

In addition, because of the one-to-one correspondence between primal and dual grid, we have that

$$H_p(\mathcal{K}) = H_p(\tilde{\mathcal{K}}), \ \ \forall p \in \mathbb{N}$$

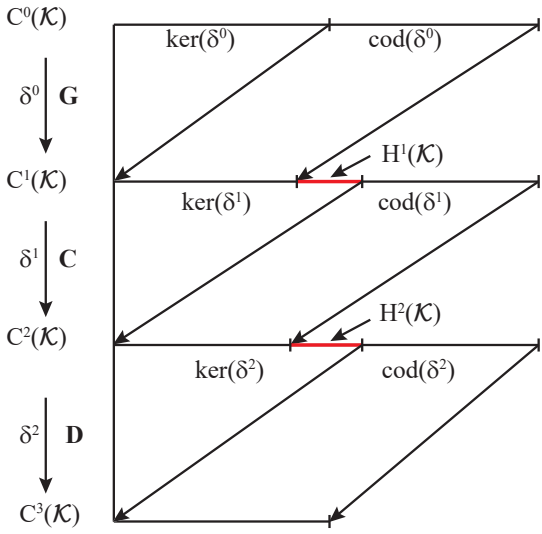

Figure 3.16: Classification of p-forms by means of the boundary operator. In red: the cocycles that are not coboundaries of any cochain.

from which it follows that

$$H^p(\mathcal{K}, \partial\mathcal{K}) \equiv H_{3-p}(\tilde{\mathcal{K}}) \tag{3.28a}$$

$$H_p(\mathcal{K}, \partial\mathcal{K}) \equiv H^{3-p}(\tilde{\mathcal{K}}). \tag{3.28b}$$

We underline that the use of the symbol "$\equiv$" justifies an exact equivalence between the two groups whereas in the previous statement the relation was just in terms of an isomorphism.

This last relations allow the application of (co)homology definitions between primal and dual complex.

4

A geometric volume integral formulation for eddy currents

So far, the discrete geometrical framework constituted by chains and cochains on a cell complex $\mathcal{K}$ led to a geometrical description of Maxwell's equations by means of global physical quantities expressed as cochains. In this chapter, after introducing the Discrete Geometrical Approach (DGA) which is a natural continuation of the previously exposed cell method, we then propose a novel volume integral formulation for eddy currents on polyhedral grids detailing the construction of the discrete counterpart of the Hodge operator based on geometrical shape functions defined in the DGA framework. These shape functions will represent the last missing brick to achieve a numerical model that can equivalently express an eddy current problem in the 3D space as a system of equations whose matrix is symmetric.

Successively, since integral formulations lead to deal with a fully populated matrix whose entries are complex numbers, we will provide some tools to work on the cyclic symmetry of the computational domain in order to reduce the size of this matrix to be assembled and stored. To face this issue, we take advantage of circuit theory, algebraic topology and of cohomology computation showing how some subtle aspects can be treated and how others can be avoided by means of them. In regard to this, we will first consider cyclic symmetry on 2D boundary integral formulation for thin conducting structures to then extend the approach to the volume integral code for solid domains.

4.1 The DGA framework

The first novelty of the integral formulation we are delineating is represented by the possibility of dealing with polyhedral grids constituted by arbitrarily shaped volume. This type of mesh can arise, for example, when subgridding is applied to a former mesh built with hexahedra or when the geometry to be discretized has narrow corners or other tricky shapes that would need too many elements if meshed with a simplicial grid.

Due to this requirement, in order to build the two missing discrete Hodge operators

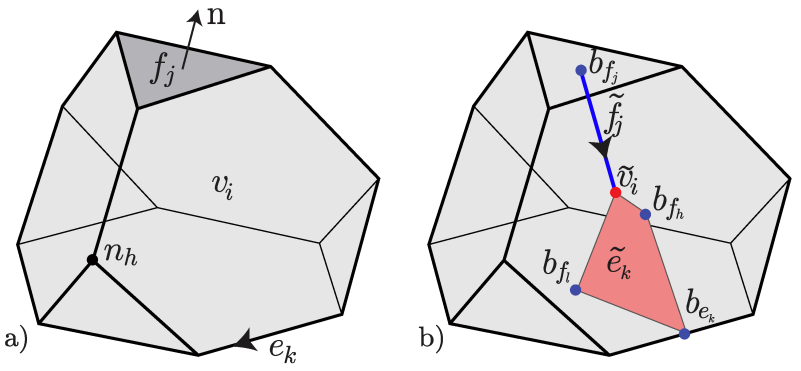

Figure 4.1: a) a polyhedron and its primal geometrical elements; b) dual barycentric partition and corresponding dual geometrical elements.

$\left[\rho^{\star}_{\tilde{f},\tilde{e}}\right]$ and $\left[\mu^{\star}_{f,\tilde{f}}\right]$, we have to provide some shape functions that can be defined on a general polyhedral volume v with an arbitrary number of faces, edges and nodes as the one reported in figure 4.1a. As a consequence, we cannot take advantage of Whitney's shape functions as in [33, 34] since they are well-defined just for tetrahedra whereas their extension to hexahedra and prisms [35] has been shown to lead to not consistent matrices [36]. For this reason, we resort to other functions, based on the *Discrete Geometrical Approach* extensively described in [37]. These basis functions, that similarly to Whitney's edge and face functions refers to the geometrical elements of the grid, are vector functions, piece-wise uniform in portions of each polyhedral volume of the grid [38, 39].

The discrete counterpart of the Hodge operator we want to achieve has to be a symmetric, positive definite (SPD) and consistent matrix. The first two characteristics deal with the stability of the method when the system solution has to be found whereas the consistency condition [40] deals with the very definition of a discrete Hodge. In fact, since a shape function is a truncated interpolation of a vector field and thus it is its approximated and discrete numerical representation, we cannot expect that the mapping between two global physical variables via the discrete Hodge is exact; on the contrary, it is asked that this approximation is exact at least for a field of the same order of the shape function truncation. In that case, the discrete operator will be consistent.

4.1.1 Shape functions

Let us consider the two interlocked grids of figure 4.1. The polyhedron on the left will be a volume v_i, $i = 1, ..., V$ of the primal grid $\mathcal{K}$ constituted by its n_h, $h = 1, ..., N$ nodes, e_k, $k = 1, ..., E$ edges and f_j, $j = 1, ..., F$ faces. Then, on the right, we have the dual of v_i on $\tilde{\mathcal{K}}$ with $\tilde{n}_h$, $h = 1, ..., N$ dual volumes, $\tilde{e}_k$, $k = 1, ..., E$ dual faces, $\tilde{f}_j$, $j = 1, ..., F$ dual edges and $\tilde{v}_i$, $i = 1, ..., V$ dual nodes. In the equations contained in the following, we will refer to an arbitrary geometric element $r_j \in \mathbf{R}^3$ whose components are $r_j = [r_j^1, r_j^2, r_j^3]^T$ which can be either a face or an edge, and to its dual one denoted $\tilde{r}_j = [\tilde{r}_j^1, \tilde{r}_j^2, \tilde{r}_j^3]^T$.

We now introduce the *double tensor* T_j

$$T_j = \tilde{r}_j \otimes r_j = \begin{bmatrix} \tilde{r}_j^1 r_j^1 & \tilde{r}_j^1 r_j^2 & \tilde{r}_j^1 r_j^3 \\ \tilde{r}_j^2 r_j^1 & \tilde{r}_j^2 r_j^2 & \tilde{r}_j^2 r_j^3 \\ \tilde{r}_j^3 r_j^1 & \tilde{r}_j^3 r_j^2 & \tilde{r}_j^3 r_j^3 \end{bmatrix}, \qquad \forall j = 1, ..., R$$

where R is the number of r-elements i.e. edges or faces of a volume v_i, and where $\otimes$ is the tensor product.

From the previous definition, trace of T follows as

$$t_j = tr(T_j) = \tilde{r}_j \cdot r_j, \qquad \forall j = 1, ..., R$$

where $[\,\cdot\,]$ is the inner product between $\tilde{r}_j$ and r_j. By right multiplying the last relation by an arbitrary vector x, from the definition of T_j,

$$T_j x = (r_j \cdot x)\tilde{r}_j \tag{4.1}$$

holds.

Property. *Due to the barycentric partition of v_i, denoting with* $\mathbf{I}$ *the identity matrix and with $|v_i|$ the measure of the volume, the identity*

$$\mathbf{T} = \sum_{j=1}^{R} T_j = \sum_{j=1}^{R} \tilde{r}_j \otimes r_j = |v_i|\mathbf{I} \tag{4.2}$$

holds[1]*.*

It follows that tensor $\mathbf{T}$ is symmetric and it is true that $tr(\mathbf{T}) = 3|v|$.

We now consider a face f_j and its dual edge $\tilde{f}_j$ of a polyhedron v_i. As shown in figure 4.2, thanks to the one-to-one correspondence between $\mathcal{K}$ and $\tilde{\mathcal{K}}$ it is possible to identify some subregions according to each *primal-dual* pair on a face f_j or on an edge e_j. In particular, since we want to map the current $\mathbf{I}$ on the primal faces with either $\tilde{\mathbf{A}}$ or $\tilde{\mathbf{U}}$ on the dual edges, we will now focus on the partition derived from the *face-dual edge* couple but we remark that similar results can be carried out also by means of the alternative *edge-dual face* partition.

In the figure, the pyramid τ_j^f is shown which has as base f_j and as apex the $\tilde{v}_j$ dual node. Similarly, the same partition can be identified considering the same pyramid $\tau_j^{\tilde{f}}$ constituted by the same dual node and by the dual edge $\tilde{f}_j$ thus $\tau_j^f \equiv \tau_j^{\tilde{f}}$ follows. With a specular reasoning, it is determined the subregion $\tau_j^e = \tau_j^{\tilde{e}}$ as a double tetrahedron whose two tetrahedra have as vertices the dual node $\tilde{v}_i$, the pair of nodes of e_j, the faces barycenters which have e_j in common and finally the barycenter of e_j.

Property. *In a subregion τ_j^r, we have that*

$$t_j = tr(T_j) = 3|\tau_j^r| \tag{4.3}$$

[1] We notice that, in this case, $\mathbf{T}$ is not the integral of the electric vector potential previously introduced.

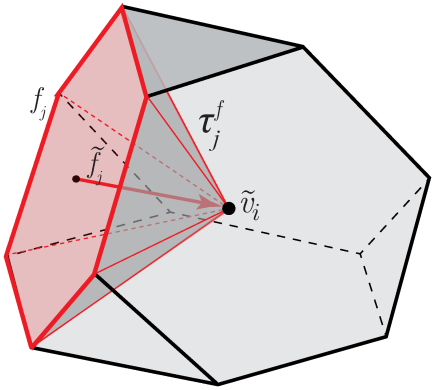

Figure 4.2: The partition $\tau_j^f \equiv \tau_j^{\tilde{f}}$ of a polyhedron v based on the one-to-one correspondence between f_j and $\tilde{f}_j$.

where $|\tau_j^r|$ is the value of the subregion volume. Proof of this and of the subsequent properties can be found in [37].

For the sake of generality, let us now consider a vector field $x(\boldsymbol{r})$ in v, such as the electric field $\boldsymbol{e}$ or the current density field $\boldsymbol{j}$. We can then compute the following integral

$$X_i^r = \int_{r_i} x(\boldsymbol{r}) \cdot d\boldsymbol{r} \tag{4.4}$$

where the arbitrary X_i^r will coincide with a global physical variable since the integral can be either considered as an integral over a face f_i or along an edge e_i. For instance, in our case it could be the integral of $\boldsymbol{j}(\boldsymbol{r}, t)$ over a face.

If $x(\boldsymbol{r})$ is constant in v, from which it follows that $X_i^r = x \cdot r_i$, by right multiplying (4.2) by x and by considering equations (4.1) and (4.4), we end up on

$$x = \frac{1}{|v|} \sum_{i=1}^{R} X_i^r \tilde{r}_i \quad i = 1, ..., R. \tag{4.5}$$

Taking into consideration the identity

$$\mathbf{I} = \frac{T_j}{t_j} + \left(\mathbf{I} - \frac{T_j}{t_j} \right) \tag{4.6}$$

by right multiplying both its members by x, thanks to (4.1), we obtain the equality

$$x = \frac{X_j^r}{t_j} \tilde{r}_j + \left(\mathbf{I} - \frac{T_j}{t_j} \right) x. \tag{4.7}$$

Finally, plugging (4.5) into (4.7), a constant field x in v can be interpolated as

$$x(\boldsymbol{r}) = \sum_{i=1}^{R} w_i^r(\boldsymbol{r}) X_i^r \tag{4.8}$$

where

$$w_i^r(\boldsymbol{r}) = \frac{\tilde{r}_j}{t_j}\delta_{ij} + \left(\mathbf{I} - \frac{T_j}{t_j}\right)\frac{\tilde{r}_i}{|v|} \tag{4.9}$$

$\forall \boldsymbol{r} \in \tau_j^r,\ \ j = 1, ..., R$. The value of the *Kronecker delta* δ_{ij} is

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

From its definition, it follows that $w_i^r(\boldsymbol{r})$ is a vector function, constant in each $\tau_j^r,\ \ \forall\, j = 1, ..., R$ and piece-wise uniform in the polyhedron v. For $w_i^r(\boldsymbol{r})$ to be a shape function and to yield consistent discrete constitutive relations, the following three properties are necessary.

Property 1. *The functions $w_i^r(\boldsymbol{r})$ are linearly independent and represent a basis. Thus,*

$$\int_{r_j} w_i^r(\boldsymbol{r}) \cdot\ dr = \delta_{ij} \tag{4.10}$$

for $i, j = 1, ..., R$ holds.

Proof. The proof can be found in [37], pp. 7405.

Property 2. *The basis vector functions exactly represent a vector field x from its degrees of freedom X_i^r at least for a uniform field.*

Proof. It directly follows from (4.8).

This last property means that the basis vector functions can exactly interpolate an $H(div)$ vector field up to a constant. In other words:

$$x = \sum_{i=1}^{R} w_i^r(\boldsymbol{r}) X_i^r + \mathcal{O}(h)$$

where h is the mesh size grain i.e. the radius of the maximum ball containing a single volume of the grid. The last expression has to be interpreted in the sense that if $h \to 0$ then the vector field will be truly constant in each infinitesimal volume and it can be exactly interpolated by means of (4.8).

Finally:

Property 3. *Consistency condition writes*

$$\int_v w_i^r(\boldsymbol{r})\, dv = \tilde{r}_i \tag{4.11}$$

with $i = 1, ..., R$.

Proof. The proof can be found in [37], pp. 7406.

The above property follows from the requirements on the discrete constitutive matrices we are going to introduce. In fact, given the shape functions interpolating the field exactly up to a certain order as shown, we want the same property also for the discrete Hodge [41, 42]. Following the reasoning in [43], this requirement for the DGA framework turns into the fulfilment of the identity

$$\frac{1}{2}\int_{v_k} x' \cdot y \, dv = \frac{1}{2}\sum_{i=1}^{R} X'_i \cdot \tilde{Y}_i \tag{4.12}$$

where x' and y are two independent vector fields, uniform in the volume v_k and where coherently $X'_i = \int_r x' \cdot dr$, $\tilde{Y}_i = \int_{\tilde{r}} y \cdot d\tilde{r}$. It can be proved that this identity is equivalent to (4.11); in fact, thanks to (4.8),

$$\frac{1}{2}\int_{v_k} x' \cdot y \, dv = \frac{1}{2}\sum_{i=1}^{R} X'_i \int_{v_k} w_i^r \cdot y \, dv = \frac{1}{2}\sum_{i=1}^{R} X'_i \cdot \tilde{Y}_i$$

that, for a uniform y, leads to

$$y \cdot \int_{v_k} w_i^r \, dv = \tilde{Y}_i$$

which is verified whenever

$$\int_{v_k} w_i^r \, dv = \tilde{r}_i. \quad \square$$

Identity (4.12) has been chosen because it conveys the essence of the discrete constitutive relations that are going to be illustrated in the continuation.

4.1.2 Discrete constitutive relation: R and M matrices

We have now to build two matrices that will be the discrete approximation of the Hodge operators. Respectively,

- $\mathbf{R} := \left[\rho^{\star}_{\tilde{f},\tilde{e}}\right]$ is the discrete counterpart of the constitutive relation between $\tilde{\mathbf{U}}$ and $\mathbf{I}$, thus it represents the resistivity tensor ρ in the computational domain;
- $\mathbf{M} := \left[\mu^{\star}_{\tilde{f},\tilde{f}}\right]$ is the discrete counterpart of the constitutive relation between $\tilde{\mathbf{A}}$ and $\mathbf{I}$, thus it represents the magnetic permeability tensor μ in the computational domain.

As already said, the two operators map a vector field integrated on the primal complex to a vector field on its dual. Thus, because of the three properties previously delineated, it seems to be worth to follow an approach based on the energy of the two fields to be mapped, as proposed in [44]. Specifically, for the magnetic permeability matrix we resort to [45] due to its peculiar double-integral form that, as we are going to see, causes $\mathbf{M}$ to be a dense matrix.

R matrix

The following pair of geometric elements is considered: $f_i \in \mathcal{K}$ and $\tilde{f}_i \in \tilde{\mathcal{K}}$ with $i = 1, ..., F$. We then recall the DOFs representing $\boldsymbol{j}(\boldsymbol{r}, t)$ and $\boldsymbol{e}(\boldsymbol{r}, t)$ in the discrete computational domain

$$I_i^f(t) = \int_{f_i} \boldsymbol{j}(t) \cdot \, ds$$

and

$$\tilde{U}_i^{\tilde{f}}(t) = \int_{\tilde{f}_i} \boldsymbol{e}(t) \cdot \, dl$$

where, if the previous hypothesis hold, $\boldsymbol{j}$ and $\boldsymbol{e}$ are uniform in the volume where the integration is performed.

$\mathbf{R}$ matrix can be achieved by starting from the expression of the dissipated power[2] in a volume v of $\mathcal{K}$:

$$\mathcal{W} = \frac{1}{2} \int_v \boldsymbol{j}' \cdot \boldsymbol{e} \, dv. \tag{4.13}$$

Since w_i^r are vector valued basis functions and if ρ is uniform in v, we can write that $\boldsymbol{j}' = \sum_{i=1}^F w_i^f I_i$ and $\boldsymbol{e} = \rho \sum_{j=1}^F w_j^f I_j$ since $\boldsymbol{e} = \rho \boldsymbol{j}$ and thanks to (4.8). Thus,

$$\mathcal{W} = \frac{1}{2} \int_v \sum_{i=1}^F w_i^f I_i \rho \sum_{i=1}^F w_j^f I_j \, dv = \frac{1}{2} \sum_{j=1}^F \left[\sum_{i=1}^F I_i \int_v w_i^f \rho w_j^f \, dv \right] I_j. \tag{4.14}$$

On the other hand, since identity (4.12) holds, it is also true that

$$\mathcal{W} = \frac{1}{2} \int_v \boldsymbol{j}' \cdot \boldsymbol{e} \, dv = \frac{1}{2} \sum_{i=1}^F I_i' \tilde{U}_i \tag{4.15}$$

where property (4.11) has been used.

In conclusion, from a comparison between (4.15) and (4.14) it follows that

$$\tilde{U}_i = \frac{1}{2} \sum_{j=1}^F \left[\int_v w_i^f \rho w_j^f \, dv \right] I_j \tag{4.16}$$

and the matrix

$$R_{ij}^{f\tilde{f}}(\rho) = \int_v w_i^f(\boldsymbol{r}) \cdot \rho w_j^f(\boldsymbol{r}) \, dv \qquad i, j = 1, ..., R \tag{4.17}$$

is the desired discrete counterpart of the constitutive relation linking $\mathbf{I}$ to $\tilde{\mathbf{U}}$.

For the sake of completeness, we here notice that the integration of (4.17) can be performed exactly since the shape functions are piece-wise uniform. Thus, for each $p_k \in \tau_k^f \;\; k = 1, ..., F$, the equation writes

$$R_{ij}^{f\tilde{f}}(\rho) = \sum_{k=1}^F w_i^f(p_k) \cdot \rho w_j^f(p_k) \frac{t_k}{3}. \tag{4.18}$$

[2]Hereafter, the time dependence of the fields is neglected for the readability of the expression

This last expression proves that $\mathbf{R}$ is sparse. Moreover, it is an SPD consistent matrix.

M matrix

We are now looking for a matrix $\mathbf{M}$ which can map $\mathbf{I}$ into $\tilde{\mathbf{A}}$.

By recalling the definition of $\tilde{\mathbf{A}}$, for a uniform $\boldsymbol{a}(\boldsymbol{r},t)$ in a volume $v_h \in \mathcal{K}$ it is possible to state that

$$\tilde{A}_i(t) = \boldsymbol{a}(t) \cdot \tilde{f}_i \tag{4.19}$$

with $\tilde{f}_j$ a dual edge in a one-to-one correspondence with a primal face f_j.

Again, by taking advantage of (4.11), the above equation also writes

$$\tilde{A}_i(t) = \int_{v_h} \boldsymbol{a}(t) \cdot w_i^f(\boldsymbol{r})\, dv, \;\; i = 1, ..., F^h \tag{4.20}$$

with F^k the number of faces of v_h.

The constitutive relationship between $\boldsymbol{j}(\boldsymbol{r},t)$ and $\boldsymbol{a}(\boldsymbol{r},t)$ is given by the *integral* relation of Biot–Savart law already exposed in the previous chapter, that reads

$$\boldsymbol{a}(\boldsymbol{r},t) = \frac{\mu_0}{4\pi} \int_v \frac{\boldsymbol{j}(\boldsymbol{r}',t)}{|\boldsymbol{r}' - \boldsymbol{r}|} dv$$

for a uniform μ_0 in Ω.

With a uniform $\boldsymbol{j}(\boldsymbol{r}',t)$ in a volume $v_k \in \mathcal{K}$, we can discretize the previous expression by means of (4.8) as

$$\boldsymbol{a}(t) = \frac{\mu_0}{4\pi} \sum_{k=1}^{V} \sum_{j=1}^{F^k} I_j^k(t) \int_{v_k} \frac{w_j^f(\boldsymbol{p}_k)}{|\boldsymbol{r} - \boldsymbol{p}_k|}\, dv. \tag{4.21}$$

where V is the total number of volumes of Ω.

If we substitute (4.21) into (4.20) we end up on the discretized expression linking $\tilde{A}_i$ to I_j that writes

$$\tilde{A}_i^h = \frac{\mu_0}{4\pi} \sum_{k=1}^{V} \sum_{j=1}^{F^k} I_j^k(t) \int_{v_h} \int_{v_k} \frac{\boldsymbol{w}_{i,h}^f(\boldsymbol{r}) \cdot \boldsymbol{w}_{j,k}^f(\boldsymbol{p})}{|\boldsymbol{r} - \boldsymbol{p}|}\, dv_{\boldsymbol{r}}\, dv_{\boldsymbol{p}} \tag{4.22}$$

which can be expressed by using the constitutive matrix $\mathbf{M}$ as

$$\tilde{A}_i^h = \sum_{k=1}^{v} \boldsymbol{M}^{hk} I_j^k. \tag{4.23}$$

In conclusion, the entries of local $\mathbf{M}^{hk}$ matrix of dimension $F^k \times F^k$, between two volume v_h and v_k are:

$$M_{ij}^{hk} = \frac{\mu_0}{4\pi} \int_{v_h} \int_{v_k} \frac{\boldsymbol{w}_i^h(\boldsymbol{r}) \cdot \boldsymbol{w}_j^k(\boldsymbol{p})}{|\boldsymbol{r} - \boldsymbol{p}|}\, dv_{\boldsymbol{r}}\, dv_{\boldsymbol{p}} \tag{4.24}$$

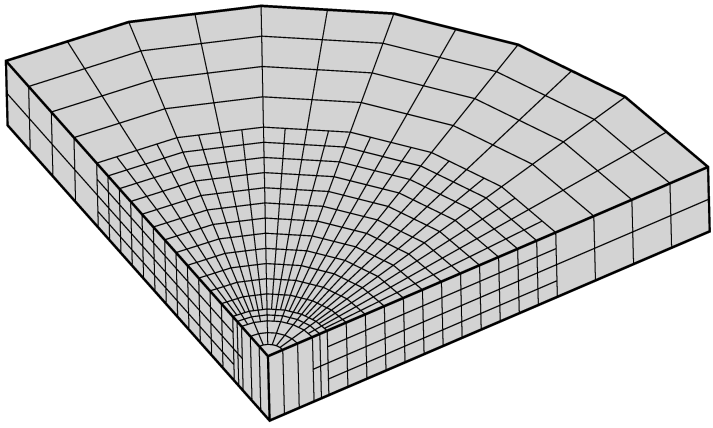

Figure 4.3: Subgridding in a polyhedral mesh of a thick slab. In the center, it can be noticed that an arbitrary polyhedron was used to adaptively improve the mesh.

Similarly to $\mathbf{R}$, also the magnetic constitutive matrix is symmetric, positive definite and consistency holds for I_j, $\tilde{A}_i$ uniform in v_k, v_h; differently from $\mathbf{R}$, $\mathbf{M}$ matrix is a *dense* constitutive matrix. In fact, from equation (4.24), M^{hk} exists for each $v_h, v_k \in \mathcal{K}$, $h, k = 1, ..., V$ thus all its entries are different from zero.

4.2 The system of equations

We are now going to present all the set of equations to solve eddy currents in the frequency domain on conductors with an arbitrary topology under the hypothesis that there is no magnetic medium outside the conductor and that the magnetic permeability of the conducting domain Ω_c is that one of the free space μ_0. In the present section, we also explain how to efficiently compute a basis for cohomology generators by means of *lazy generators* [46, 47] belonging to the first cohomology group $H^1(\partial\mathcal{K}, \mathbb{Z})$: since the computation of these generators if performed in a wrong way could lead to have an unacceptable bottleneck in the overall simulation, a particular attention will be dedicated to them. Moreover, as exposed, we resort to piece-wise uniform basis functions defined in the DGA framework in order to treat meshes composed by arbitrary polyhedra. In the past few years polyhedral grids have become particularly appealing for subgridding to obtain adaptive mesh refinement and de-refinement [48] thus the possibility of dealing with polyhedra represent an interesting feature for a numerical code. An example of this kind of mesh is shown in figure 4.3.

All these features are encapsulated into a new geometric volume integral code whose details are described in the following.

4.2.1 Boundary conditions and gauging

Before proceeding with the dissertation about non simply connected domains, we now pause on the choice of boundary conditions (BCs) and on the solution gauging. Yet,

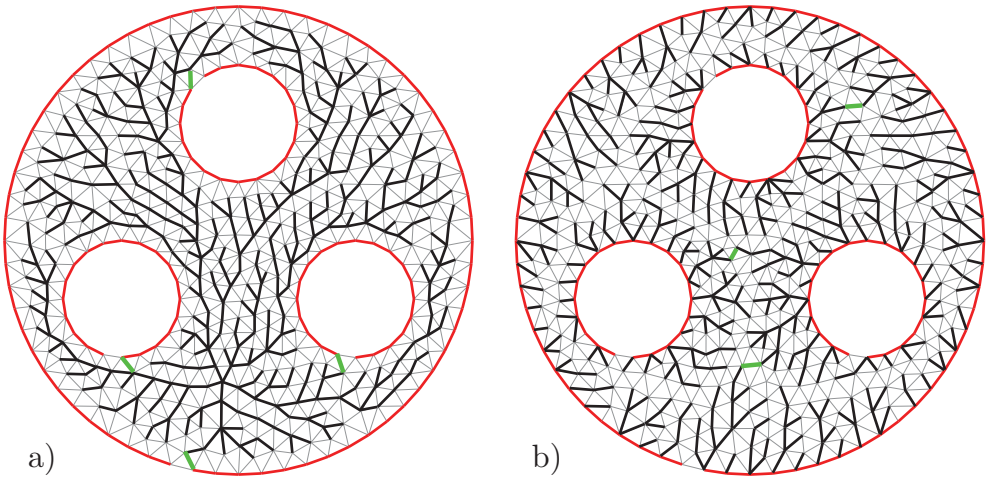

Figure 4.4: A manifold with three cavities. a) The gauge is obtained by joining (green segments) the C distinct trees on each connected components of $\partial\mathcal{K}$ to the tree in $\mathcal{K}\setminus\partial\mathcal{K}$; b) C distinct trees on each connected components of $\partial\mathcal{K}$ are expanded into $\mathcal{K}\setminus\partial\mathcal{K}$ and then joined each other (green segments).

these two points are strictly related as explained in [49] and this leads to a unique approach that can address both the issues.

Starting from the solution gauging, it is a well known fact that since $\boldsymbol{j} = \nabla \times \boldsymbol{t}$ the solution in terms of $\boldsymbol{t}$ is not unique. In fact, if we denote as $\mathbf{F}$ the array of the degrees of freedom, this is determined except for the gradient of a scalar potential. If this potential $\boldsymbol{\Omega}$ is introduced we thus have

$$\mathbf{F} = \mathbf{T} + \mathbf{G}\boldsymbol{\Omega}$$

which in any case yields $\mathbf{I} = \mathbf{C}\mathbf{F} = \mathbf{C}\mathbf{T}$. If we want to convey the uniqueness of the DOFs of the problem we have to set $\mathbf{G}\boldsymbol{\Omega}$ to zero: this can be done by using a tree-cotree technique and by imposing the values of $\mathbf{T}$ on the edges belonging to a tree of $\mathcal{K}$ equal to zero as proposed in [50, 51].

Furthermore, also boundary conditions for this volume integral formulation can be imposed by means of a tree-cotree decomposition on the edges of $\partial\mathcal{K}$. In fact, since we want to impose that no current can flow outside the conducting domain Ω_c then by building a tree on $\partial\mathcal{K}$ and by setting to zero its elements, thanks to a circuital interpretation, we obtain that each free DOF on an edge that belongs to the cotree will close a loop on $\partial\mathcal{K}$ thus imposing the value of the current. Since $\boldsymbol{j}\cdot\boldsymbol{n} = 0$ on $\partial\Omega$ is required, it follows that the DOFs values on the edges of the cotree have to be set to zero. In conclusion, boundary conditions on the current direction can be determined by simply annihilating all the values of $\mathbf{F}$ on $\partial\mathcal{K}$.

The technique to achieve the desired boundary conditions also reflects back on the DOFs gauging. In fact, if we compute a tree in the whole $\mathcal{K}$, it can happen that if an edge of the cotree belongs to $\partial\mathcal{K}$ then the current flowing across the closed surface composed by the cotree edge and the other tree edges is zero since all the DOFs on the

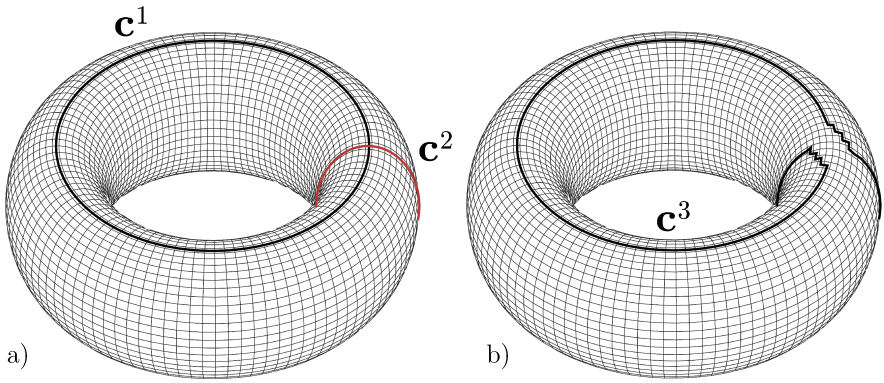

Figure 4.5: a) The support of two distinct representative of $H^1(\partial\mathcal{K})$. b)The support of a mixed representative of $H^1(\partial\mathcal{K})$ as a linear combination of $\boldsymbol{c}^1$ and $\boldsymbol{c}^2$.

domain boundary were set to zero because of the BCs. To prevent this situation, the computation of the tree for the solution gauging has to extend from $\partial\mathcal{K}$ toward $\mathcal{K} \setminus \partial\mathcal{K}$.

From the implementation point of view, two different approaches are conceivable:

- the first approach, represented in figure 4.4a and implemented in [3], prescribes the construction of C distinct trees on each connected components of $\partial\mathcal{K}$ and then another independent tree that belongs to $\mathcal{K} \setminus \partial\mathcal{K}$. Once that all these trees are identified thanks to a *breadth first search* (BFS, see [52]) the trees on $\partial\mathcal{K}$ are joined to the inner tree by adding C edges of the cotree of $\mathcal{K} \setminus \partial\mathcal{K}$. Successively, all the ith entries of $\mathbf{T}$ related to this tree are set to zero;

- the other possible technique, suggested in [53] and then here applied to a volume integral code for the first time, is based on the construction of C distinct trees for each connected components of $\partial\mathcal{K}$ that are successively expanded inside the computational domain as shown in figure 4.4b. Later, $(C-1)$ edges of the cotree are added to build a unique tree and then the related $\mathbf{T}$ values are annihilated.

Even if in [53] it is proved that the two approaches are equivalent, in the present work the second one is chosen since it does not lead to any shortcoming when cyclic symmetry has to be taken into account in the computation as it will be shown in the continuation.

As final remark, we here state that figure 4.4 and all the figures containing simplicial meshes in the continuation have been created thanks to *Gmsh* as a mesh generator [54].

4.2.2 Efficient (co)homology computation

By continuing the dissertation contained in section 3.4, we now have to take into consideration the more general case of a non simply connected domain also referred to as *non-trivial* domain. If we focus on the torus in figure 4.5a and on the degrees of freedom of the problem, we can conclude that, with respect to the solenoidal currents $\mathbf{I}$ that flow through the faces of a trivial domain, some additional non-local independent currents will act, whose basis is not expressed by means of the topological incidence matrices

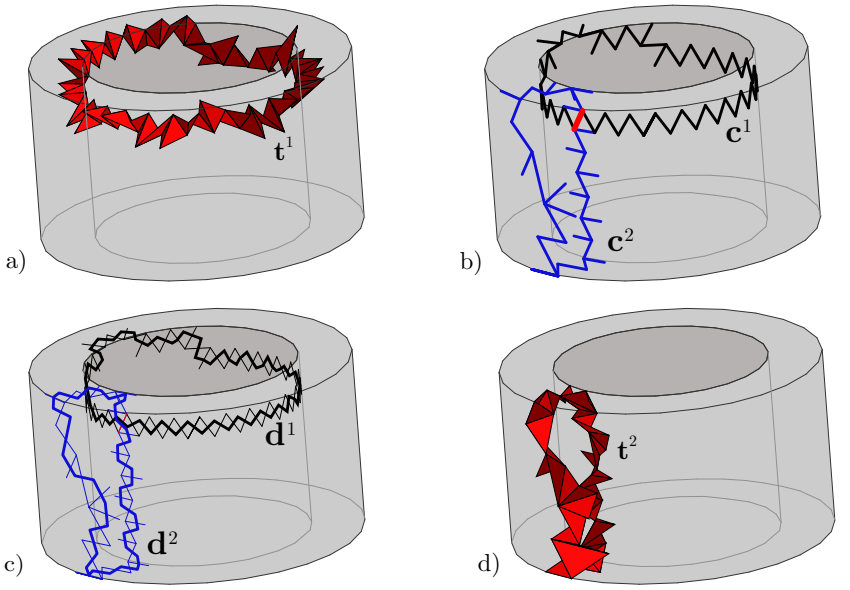

Figure 4.6: Examples of cohomology generators $H^2(\mathcal{K}, \partial\mathcal{K})$ and $H^1(\partial\mathcal{K})$ for a solid torus. a) The support of a representative $\mathbf{t}^1 \in H^2(\mathcal{K}, \partial\mathcal{K})$ generator. b) The support of two representatives $\mathbf{c}^1, \mathbf{c}^2 \in H^1(\partial\mathcal{K})$ generators. c) Each $\mathbf{d}^i$, $i \in \{1, 2\}$ is the cycle made of dual edges which are dual to $\mathbf{c}^i$ in $\partial\mathcal{K}$. d) $\mathbf{t}^2 = \mathbf{C}\mathbf{c}^2$ is trivial in $H^2(\mathcal{K}, \partial\mathcal{K})$.

because of the boundary conditions imposed[3]. For this reason, if we denote with $\mathbf{i}$ these independent currents, we can write that on an arbitrary conducting domain Ω_c

$$\mathbf{I} = \mathbf{C}\mathbf{T} + \mathbf{W}\mathbf{i}. \tag{4.25}$$

We now resort to the theory related to de Rham complex [55], which demonstrates that, if the columns of $\mathbf{W}$ contain a basis of the second relative cohomology group $H^2(\mathcal{K}, \partial\mathcal{K})$, the field reconstructed through the face basis functions w^f is a vectorial field which spans fields tangent to $\partial\tilde{\mathcal{K}}$ that are solenoidal but not irrotational; indeed, this is the field we are looking for. Thus, the support $\mathbf{t}^1$ of one generator of $\mathbf{W}$ can be represented as a subset of faces crossed by a unitary non-local current as shown in figure 4.6a. Unfortunately, retrieving this basis is not efficient since we should span all the mesh faces and it does not exist any fast[4] and robust algorithm able to perform this task; alternatively, by recalling (3.28a), we have

$$H^2(\mathcal{K}, \partial\mathcal{K}) \equiv H_1(\tilde{\mathcal{K}}).$$

[3]For instance, the circulation of $\mathbf{T}$ on the cocycle $\boldsymbol{c}^2$ in figure 4.5a could express such non-local current but all the DOFs on $\partial\mathcal{K}$ have to be set to zero because of the BCs.

[4]It is here intended that *fast* means that the algorithm runs in linear time.

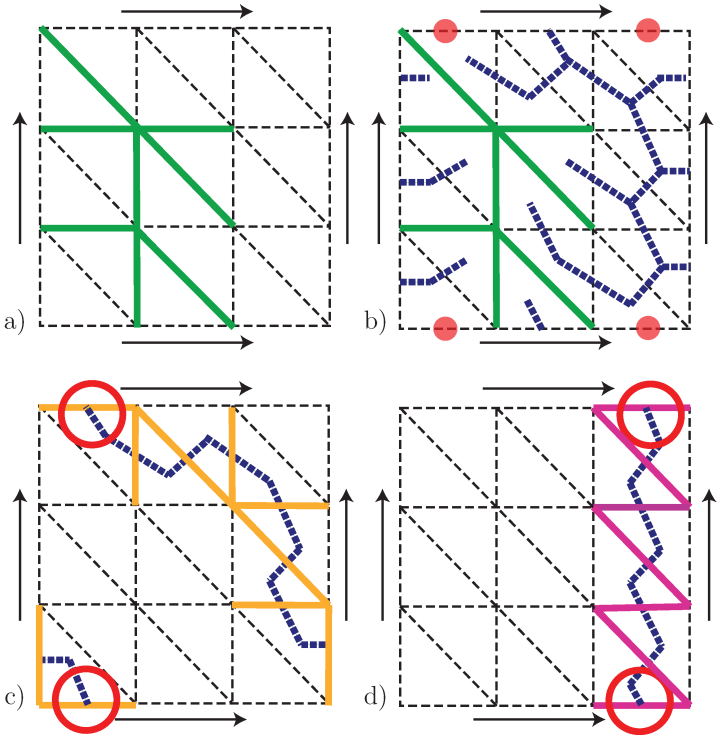

Figure 4.7: Lazy generators retrieval on a triangulation of a torus whose opposite sides are identified by the arrows. a) A tree (in green) is built on the primal grid nodes. b) A second tree (in blue) is retrieved starting from the dual nodes of the dual grid of $\partial\mathcal{K}$. Two free edges not belonging either to the primal or the dual tree are identified. c) The support of the first $H^1(\partial\mathcal{K})$ generator is identified by the path closed by the first edge not belonging to any of the previously computed trees. d) The other edge forms the second cycle that identifies the second generator.

Since a generator of $H_1(\tilde{\mathcal{K}})$ is homologous to a generator of $H_1(\partial\tilde{\mathcal{K}})$, we can finally obtain an equivalent way to express $\mathbf{t}^1$ as $\mathbf{t}^1 = \mathbf{C}\mathbf{c}^1$ where $\mathbf{c}^1 \in H^1(\partial\mathcal{K})$ thus limiting the algorithm search to the primal edges of $\partial\mathcal{K}$. Yet, this process has a drawback since for each independent $\mathbf{t}^i$ to be found we obtain two $\mathbf{c}^j$ as those represented in figure 4.6b; among these generators, the dual cycle[5] of one $\mathbf{c}^j$ of the two is trivial in the conductor $\mathcal{K}$ as $\mathbf{d}^2$ in figure 4.6c whereas the other dual ($\mathbf{d}^1$ in figure 4.6c) is trivial in the conductor complement $\mathbb{R}^3 \setminus \mathcal{K}$. The first one does not express a basis of $H^2(\mathcal{K}, \partial\mathcal{K})$ and it should be discarded whereas the second is the generator we were interested in.

One may hope to retrieve just the independent generators or at least to identify the right ones and indeed in literature many alternative ways are explored. In past, the proposed techniques were based on linear algebra as it happens in [17, 3, 56], then, since

[5]Here the reasoning is limited to $\partial\mathcal{K}$ thus its dual $\partial\tilde{\mathcal{K}}$ has to be built on the boundary faces only.

algebraic approaches may results to be somehow time consuming, the attention has been captured by combinatorial algorithms based on the topological properties of the computational domain. For instance, a first proposal can be found in [57] where a mixed algebraic-combinatorial method is described whereas, later, the computation relied on purely combinatorial methods that can run in linear time, as that one proposed in the seminal work of R. Hiptmair [58] in which, more than the generators disentanglement also the issue of the necessary change of basis is tackled, for the *homology* computation. More recently, fast combinatorial methods have been extended to *cohomology* computation too, as in [59] wherein, however, no change of basis is proposed and the disentanglement is still faced with less efficient algebraic processes or as in [60], in which the combinatorial cohomology computation is equipped with both the identification of the minimum set of generators and the change of basis. About this last aspect, we point out that a change of basis becomes necessary as soon as the retrieved generators result to be "mixed" together, as shown in figure 4.5b where the cocycle $\boldsymbol{c}^3$ results to be a linear combination of the two basis $\boldsymbol{c}^1, \boldsymbol{c}^2$. This event makes very risky a disentanglement based on the assumption that the independent generators are half of the total $H^1(\partial\mathcal{K})$ generators since when the handles m of $\mathcal{K}$ are more than one severe errors may occur.

As an additional alternative choice, in [46] it is proved that a solution keeping all the retrieved generators is also possible. This approach which is the one used for this code, is based on the so called *lazy generators*. When applied to problems in the frequency domain, as in our case, it does not lead to any degradation of the performances during the solution of the system and during the computation of current density distribution: this last is shown to be the same up to the linear solver tolerance with both the approaches that exploits either the lazy generators or just the independent ones. Moreover, with this choice, the pre-processing takes just some seconds even when the generators are numerous since the combinatorial algorithms extensively exposed in [47] and here implemented, exhibits a linear worst-case complexity.

Algorithm 1 $H^1(\partial\mathcal{K})$ basis generation

Input: $\mathbf{C}_b, \mathbf{G}_b$
Output: $\mathbf{H}^1, ..., \mathbf{H}^{2m}$ generators
procedure GeneratorsRetrieval
PrimalTree ← *FindPrimalTree(*$\mathbf{G}_b$*);*
DualTree ← *FindDualTree(*$\mathbf{C}_b$*, PrimalTree);*
[Dist, Parent] ← *FindBFSdistance(*$\mathbf{C}_b$*, DualTree);*
$g \leftarrow 0$
for each edge e_g of $\partial\mathcal{K}$ **do**
if *PrimalTree(*e_g*)=false* and *DualTree(*e_g*)=false* **then**
$l \leftarrow g + 1$
$\mathbf{H}^g$ ← *FindCocycleFast(*$\mathbf{C}_b$*, DualTree,* e_g*);*
return $\mathbf{H}^1, ..., \mathbf{H}^{2m}$

Lazy generators construction can be summarized with the pseudocode illustrated in algorithm 1. The main steps in order to retrieve all the $2m$ basis (dependent and independent) of the H^1 generators are:

- find a tree on the primal nodes of $\partial\mathcal{K}$ by exploiting the incidence matrix $\mathbf{G}_b$. This

matrix is defined as $\mathbf{G}$ matrix reduced to the only nodes belonging to the cell complex boundary;

- repeat the procedure for the dual nodes of $\partial\mathcal{K}$ so that the dual tree to be obtained does not intersect the primal tree edges. The reduced matrix $\mathbf{C}_b$ related to the only edges of $\partial\mathcal{K}$ is provided;
- starting from the dual tree root, compute the distance field $Dist$ of each node of the dual tree from the common root and store each node parent into $Parent$;
- for each primal edge $e_g \in \partial\mathcal{K}$ not belonging either to the primal or the dual tree previously obtained, find the dual edges in the dual tree that close a cycle in $\in \partial\mathcal{K}$;
- store the support of each generator into the matrix $\mathbf{H}(:, g),\ g = 1, ..., 2m$.

Some of these steps are also drawn in figure 4.7; further details and the implemented source code of each routine are reported in appendix B.

4.2.3 Equations

Once the solution gauging has been imposed, the DOFs of the problem coincide with $\mathbf{T}$, i.e. $\mathbf{F} = \mathbf{T}$. Then, if the $2m$ representative of $H^1(\partial\mathcal{K}, \mathbb{Z})$ are stored in each column of the matrix $\mathbf{H}$ as explained above, the integral of the current density over the faces of $\mathcal{K}$ reads

$$\mathbf{I} = \mathbf{CT} + \mathbf{CHi}. \tag{4.26}$$

If we make a distinction between the unknown magnetic vector potential $\tilde{\mathbf{A}}$ due to eddy currents and the already known one $\tilde{\mathbf{A}}_s$ generated by a fixed source of magnetic field as a coil so that $\tilde{\mathbf{A}}_{tot} = \tilde{\mathbf{A}} + \tilde{\mathbf{A}}_s$, we can substitute the two discrete counterpart of the constitutive relations

$$\tilde{\mathbf{U}} = \mathbf{RI} \tag{4.27a}$$

$$\tilde{\mathbf{A}} = \mathbf{MI} \tag{4.27b}$$

into (3.24) which **locally** expresses Faraday–Neumann law and by recalling equation (3.19), the following system of equation can be obtained

$$\mathbf{C}^T\mathbf{RI} + i\omega\mathbf{C}^T\mathbf{MI} = -i\omega\mathbf{C}^T\tilde{\mathbf{A}}_s. \tag{4.28}$$

where i is the imaginary unit, and $\omega = 2\pi f$ is the angular frequency with f the frequency of the source generating $\tilde{\mathbf{A}}_s$.

In addition to local Faraday's equation, also the non-local one has to be enforced as

$$\mathbf{H^T}(\mathbf{C}^T\tilde{\mathbf{U}} + i\omega\tilde{\mathbf{\Phi}}) = \mathbf{0}. \tag{4.29}$$

Thus, taking advantage of (4.26), the following system of equation is obtained:

$$\begin{bmatrix} \mathbf{K} & \mathbf{KH} \\ \mathbf{H}^T\mathbf{K} & \mathbf{H}^T\mathbf{KH} \end{bmatrix} \begin{bmatrix} \mathbf{T} \\ \mathbf{i} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_s \\ \mathbf{H}^T\mathbf{b}_s \end{bmatrix} \tag{4.30}$$

where

$$\mathbf{K} := \mathbf{C}^T \left(\mathbf{R} + i\omega\mathbf{M}\right) \mathbf{C}$$

and

$$\mathbf{b}_s := -i\omega\mathbf{C}^T \tilde{\mathbf{A}}_s.$$

4.2.4 Solution post processing and numerical results

The geometric volume integral code for non-trivial polyhedral grids has been tested on three different geometries to be validated in the frequency domain. Specifically, the employed test problems are:

1. A solid sphere (radius $a = 50\,\text{mm}$, resistivity $\rho = 0.017\,\mu\Omega\text{m}$) immersed in a uniform magnetic induction field vertically directed along the z-coordinate ($B_z = 1\,\text{T}$, $f = 50\,\text{Hz}$).
2. A thick disk ($\rho = 0.017\,\mu\Omega\text{m}$) with one hole (or one handle thus, $2m = 2$), excited by a stranded coil crossed by an imposed sinusoidal current I_c ($I_c = 1000\,\text{At}$, $f = 1\,\text{kHz}$).
3. A thick plate ($\rho = 0.017\,\mu\Omega\text{m}$) with four holes (or four handles thus, $2m = 8$), excited by a stranded coil crossed by an imposed sinusoidal current I_c ($I_c = 1000\,\text{At}$, $f = 50\,\text{Hz}$).

In order to check the solution accuracy, we compare both the computed current density $\mathbf{J}$ and the magnetic induction field $\mathbf{B}$ produced by $\mathbf{J}$ versus a reference that can be analytical for the sphere tests or numerical in the other two cases. Thus, the result obtained in terms of $\mathbf{T}$ and $\mathbf{i}$ values from the system solution has been post processed by using equation (4.5) to reconstruct the current density field inside a given volume v of the mesh as

$$J_v = \frac{1}{|v|} \sum_{j=1}^{F_v} I_j \tilde{f}_j. \tag{4.31}$$

The current I_j over the jth face is a correspondent entry of the array $\mathbf{I}$ that can be easily computed as $\mathbf{I} = \mathbf{CT} + \mathbf{CHi}$ whereas F_v is the number of faces of v. Successively, the magnetic induction field is derived from the current density values by using the closed-formulas contained in [61] which are based on simple operations between the geometrical elements of a mesh volume. For the sake of completeness these formulas are also reported in the appendix A of this work.

Solid sphere tests. Three different polyhedral grids have been adopted for the problem of the solid sphere with three progressively denser mesh grains whose details are reported in table 4.1. Moreover, in figure 4.8 and 4.9 a representation of the types of polyhedra inside each sphere grid can be found.

In figure 4.10, a representation of the real part of the current density distribution computed on the coarse mesh is showcased. In figure 4.11, the results of the computation of the magnetic induction field B_z for the three grids are compared to the analytical solution of the problem which is here used as reference. For test points that lay outside the sphere, an excellent agreement is found for each tested mesh whereas due to the

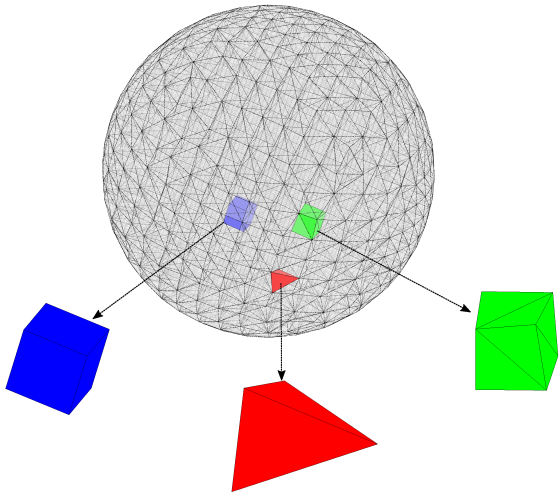

Figure 4.8: Example of three different polyhedra inside the sphere mesh. Left to right: an hexahedron (blue), a tetrahedron (red) and a polyhedron (green) with eleven faces.

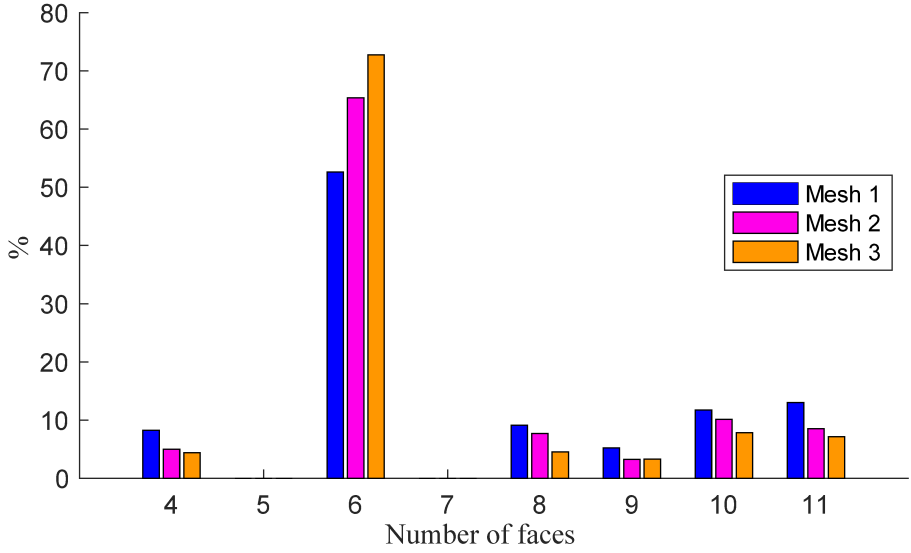

Figure 4.9: Percentage of polyhedra distinguished by the number of faces in the three sphere meshes.

	n_n	n_e	n_f	n_v	DOFs
Mesh 1	1911	7326	7256	1840	4329
Mesh 2	6055	22014	21872	5912	13505
Mesh 3	14027	49110	48860	13776	30661

Table 4.1: Sphere meshes: comparison between the number of nodes (n_n), edges (n_e), faces (n_f), volumes (n_v) and degrees of freedom.

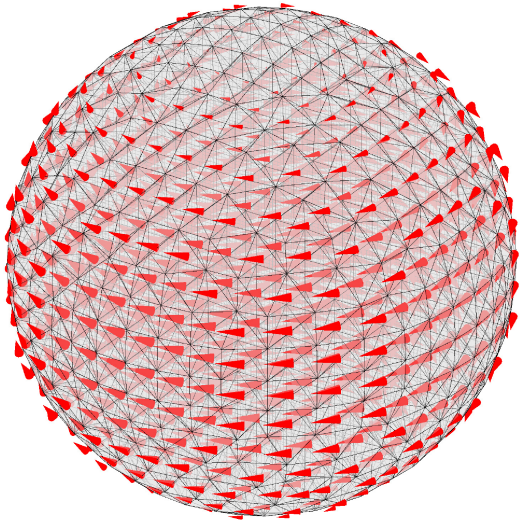

Figure 4.10: Real part of the current density field on the coarsest mesh.

mesh coarseness inside the sphere, the discrepancy (ΔB_z) for mesh 1 is much higher than either mesh 2 or mesh 3.

Thick disk test. The geometry on which test 2 has been performed is reported in figure 4.12. Since the geometry is not simply connected, lazy cohomology generators have been computed in this case: the numerical solution obtained with the lazy cohomology basis (i.e. with both the dependent and independent generators) is the same as the one with the standard cohomology basis up to the linear solver tolerance as expected. Also in this case, three discretization levels have been applied to obtain three different meshes. Since an analytical solution of the problem is not available, the numerical results during this test are compared to a 2D axisymmetric solution used as a reference whose representation can be found in figure 4.13. As far as the solution accuracy is concerned, in the same figure we notice that good results can be obtained also with a relatively coarse mesh (40×5 elements on the cross section and 18 subdivisions along the toroidal direction).

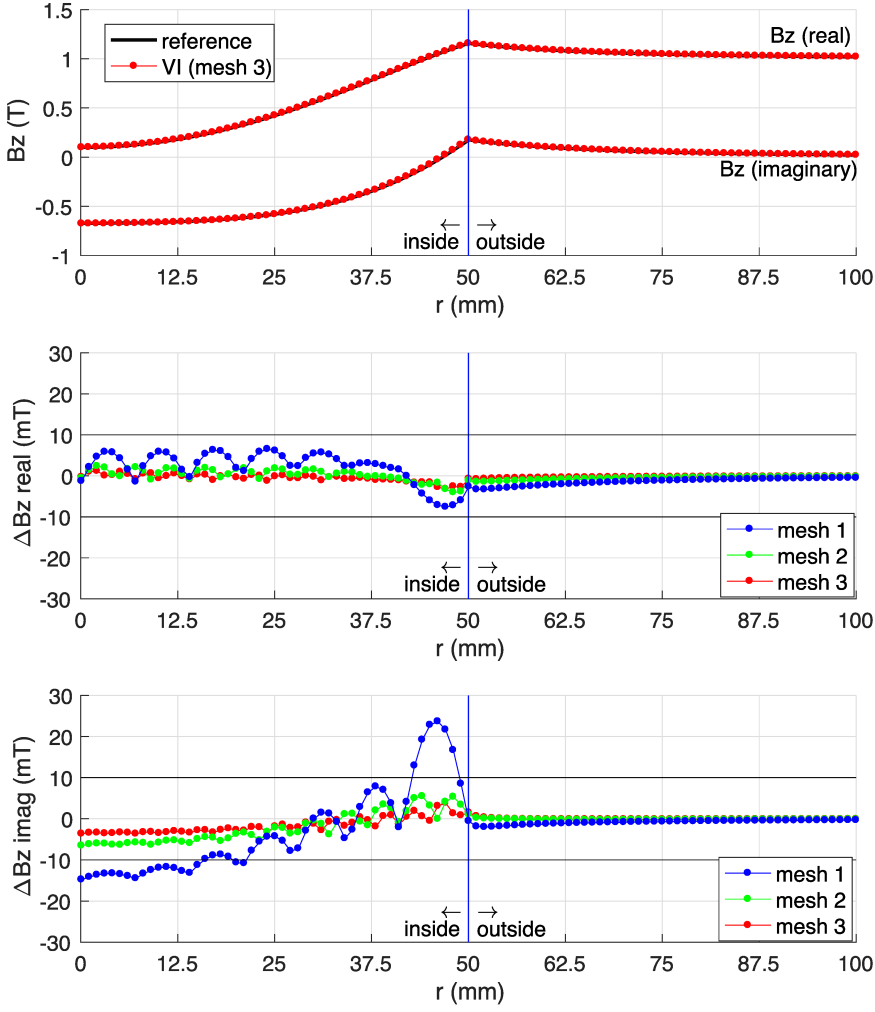

Figure 4.11: Top: comparison of the z component of the magnetic induction field (B_z) along a radial direction between the volume integral (VI) code and the analytical reference solution on the denser mesh. Middle and bottom: real and imaginary components of the B_z difference between VI and the reference for the three grids.

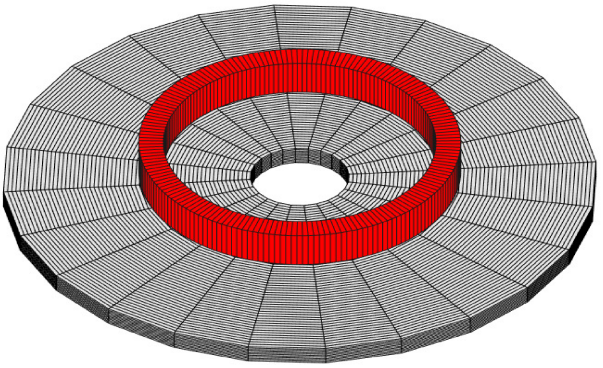

Figure 4.12: The thick disk geometry. In red: the stranded coil used as source of magnetic field.

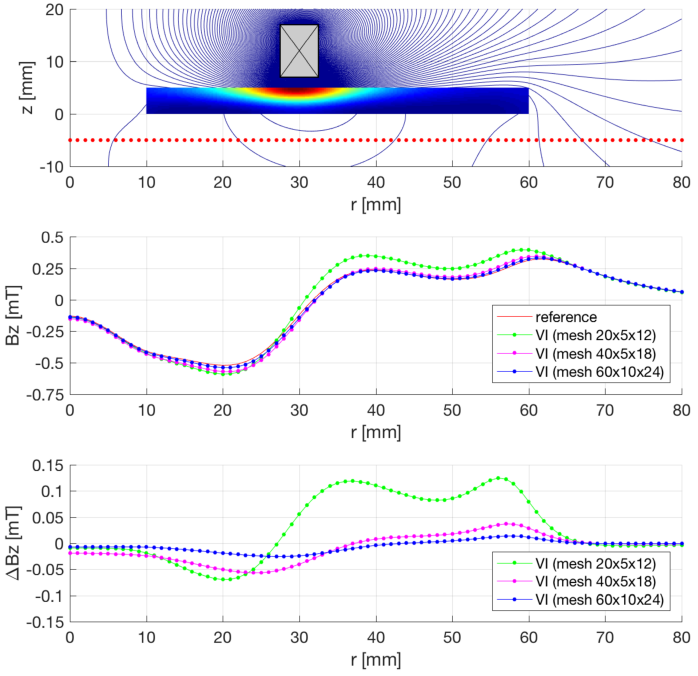

Figure 4.13: Top: magnetic field map of the 2D axisymmetric solution (reference). Middle: VI results on the computation of B_z field; for each tested mesh density, the real component of B_z ($re\{B_z\}$) field in correspondence of the red dots of the top figure is plotted. Bottom: $re\{B_z\}$ discrepancy between the volume integral code and the reference solution.

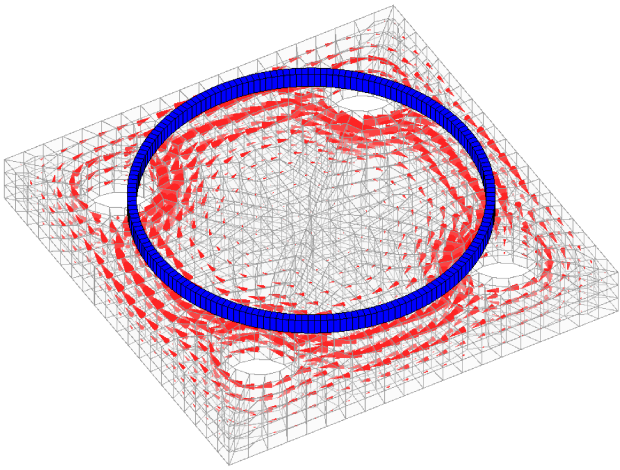

Figure 4.14: Real current density distribution for test 3. In blue: the stranded coil used as source.

Thick plate test. Least but not last, the real part of the current density ($re\{J\}$) for the geometry used in test 3 is shown in figure 4.14. The solution obtained with VI was compared to the one provided by the the 3d FEM code Computer Aided Fusion Engineering (CAFE) [62]. Again, the solution with the four independent cohomology generator is the same as the one obtained with the eight lazy cohomology basis. The dual of the support of the eight $H^1(\partial\mathcal{K}, \mathbb{Z})$ generators is shown in figure 4.15.

4.3 Dealing with cyclic symmetry

Using cyclic symmetry in order to sensibly reduce the number of DOFs of the problem to be studied is a technique that has already been widely exploited as an effective way to largely reduce the simulation time as explained for example in [63]. As a consequence, many papers have been published on how to treat symmetric domains by using different numerical formulations like in [63, 64] for finite elements method and the boundary element formulation written by means of a magnetic vector potential. Nevertheless, the topological preprocessing necessary to deal with integral formulations and symmetric non-trivial domains is still an open topic since the exploitation of cyclic symmetry adds some further aspects to the topics already proposed in the previous chapter.

To that end, we will detail all these aspects starting from the exploitation of cyclic symmetry applied to the boundary integral formulation of [65] with which we will address the issue related to the cohomology computation on a symmetric subregion of the computational domain Ω_c; later, we will extend the proposed approach to the volume integral formulation previously introduced, focusing on how to tweak the cohomology computation for a polyhedral mesh where subgridding has been applied. Then, when cyclic symmetry is applied to a volume integral formulation also some gauging-related

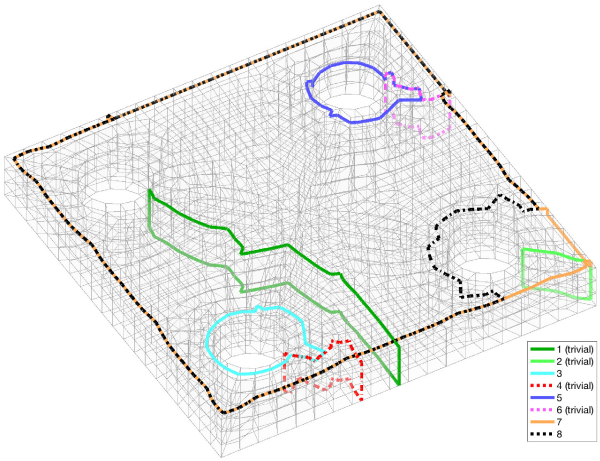

Figure 4.15: Dual of the support of the $H^1(\partial\mathcal{K}, \mathbb{Z})$ generators for test 3 geometry. Half of the generators are trivial in $\mathcal{K}$ (n. 1, 2, 4, 6), the others are non-trivial.

problems arise that have to be carefully tackled.

4.3.1 Boundary integral formulation for eddy current problems

In presence of a magnetic field whose variation in the time is slow compared to the phenomena to be studied in the problem, eddy currents are induced and start flowing in a conducting domain Ω_c. Furthermore, in a non-magnetic medium of permeability μ_0, if the frequency of the magnetic source and the resistivity ρ of the conductor are so that the penetration depth of the field

$$\delta = \sqrt{\frac{\rho}{\pi f \mu_0}} \tag{4.32}$$

is comparable to the thickness t of the conductor then the current flowing in Ω_c can be considered uniform along the thickness and tangential to a surface describing the other two dimensions of the domain. In this setting, eddy currents can be efficiently calculated by using Boundary Integral (BI) methods [66, 67, 68, 65] that allow to treat the 3D geometry as a 2D surface thus simplifying the domain representation and thus reducing the DOFs of the problem.

Similarly to volume integral methods, also boundary integral methods are based on non-local equations expressing constitutive relations as an integral on the whole computational domain which in this case is represented by the surface Σ. Following a specular procedure as the one pursued in the preceding part and by resorting to the DGA framework, also the eddy current problem on Σ, whose details are delineated in [65], leads to deal with a dense matrix. For this reason a limitation of the computational domain to a symmetry cell only instead of the overall geometry represents an important enhancement.

From the volume integral formulation to a stream function

Differently to [64] where equations are written in terms of magnetic vector potential, the BI formulation in [66] takes advantage of a nodal magnetic **scalar** potential which in literature is also referred to as a *stream function* because of the physical analogy with two-dimensional fluid flow problems. This choice, that is the one implemented in the BI formulation under analysis, leads to dealing with linear systems characterized by the minimum amount of DOFs; notwithstanding, the reduction of a 3D problem into a 2D computational domain represented by Σ commonly leads to multiply connected manifolds that make the stream function ill-defined, an aspect that in [65] is automatically treated thanks to cohomology theory.

The BI formulation which we are discussing about is a simplification of the VI code based on the fact that, since the current is uniform along the thickness t, only a single layer of prisms is necessary to discretize the 3D domain. This assumption, in addition to the prescribed boundary conditions on $\partial\mathcal{K}$ allows to consider only the vertical DOFs of the 3D geometry depicted in red in figure 4.16b since the horizontal T_i (in blue) are set to zero because of the BCs, indeed. Furthermore, since the direction of the *active* DOFs in red is known from the principle, the only left unknowns are represented by the scalar values of each of this active DOFs. Thus, in conclusion, the 3D geometry of figure 4.16a can be reduced to the one represented in figure 4.16c of which the DOFs are just nodal scalar values of a stream function, denoted in figure 4.16d and in the continuation as Ψ, and where the current $\mathbf{I}$ can be thus computed per unit of thickness.

We can now focus on the surface Σ only, and discretize it with a simplicial mesh constituted by triangles. This partition can be encoded into a 2D cell complex $\mathcal{K}$ which will be formed by N nodes $n_i,\ i = 1, ..., N$, E edges $e_j,\ j = 1, ..., E$ and F triangles $f_k,\ k = 1, ..., F$. Then, from $\mathcal{K}$ we can build a barycentric partition thus obtaining the dual nodes $\tilde{n}_k,\ k = 1, ..., F$ in a one-to-one correspondence with the primal triangles f, the dual edges $\tilde{e}_j,\ j = 1, ..., E$ in a one-to-one correspondence with the primal edges e and dual faces $\tilde{n}_i,\ i = 1, ..., N$ in a one-to-one correspondence with the primal nodes n [6]. All these dual elements belong to $\tilde{\mathcal{K}}$ and both $\mathcal{K}$ and $\tilde{\mathcal{K}}$ are represented in figure 4.16c. Also for this 2D cell complex the incidences between p-cells are encoded into the incidence matrices $\mathbf{G}$ and $\mathbf{C}$ for the primal complex and $\tilde{\mathbf{G}}$, $\tilde{\mathbf{C}}$ for $\tilde{\mathcal{K}}$.

Denoting the array of the current flowing in Ω_c expressed per unit of thickness as $\mathbf{I}$, we can state that

$$\mathbf{I} = \mathbf{G}\boldsymbol{\Psi} + \mathbf{H}\mathbf{i}, \tag{4.33}$$

where $\boldsymbol{\Psi}$ is the array of the DOFs that contains the scalar values of the stream function on the mesh nodes whereas $\mathbf{i}$ is again the array of independent currents that flow along non-local paths when the domain is not simply connected (or *non-trivial*); the columns of $\mathbf{H}$ store a set of representatives of generators of $H^1(\mathcal{K}, \partial\mathcal{K})$. We remark that due to the fact that we are now dealing with a 2D simplicial geometry, the cohomology computation does not lead to have redundant generators as in the 3D case with the *lazy generators* approach. On the contrary, we now have a bounded surface whose boundary has to be taken into account during the cohomology computation as it is done

[6] We notice that the choice on the notation here proposed for each primal-dual couple is different from the one used for the volume integral formulation. This change in the notation aim at preserving the correspondence between primal and dual edges e and $\tilde{e}$ that in a 2D mesh are numerically even.

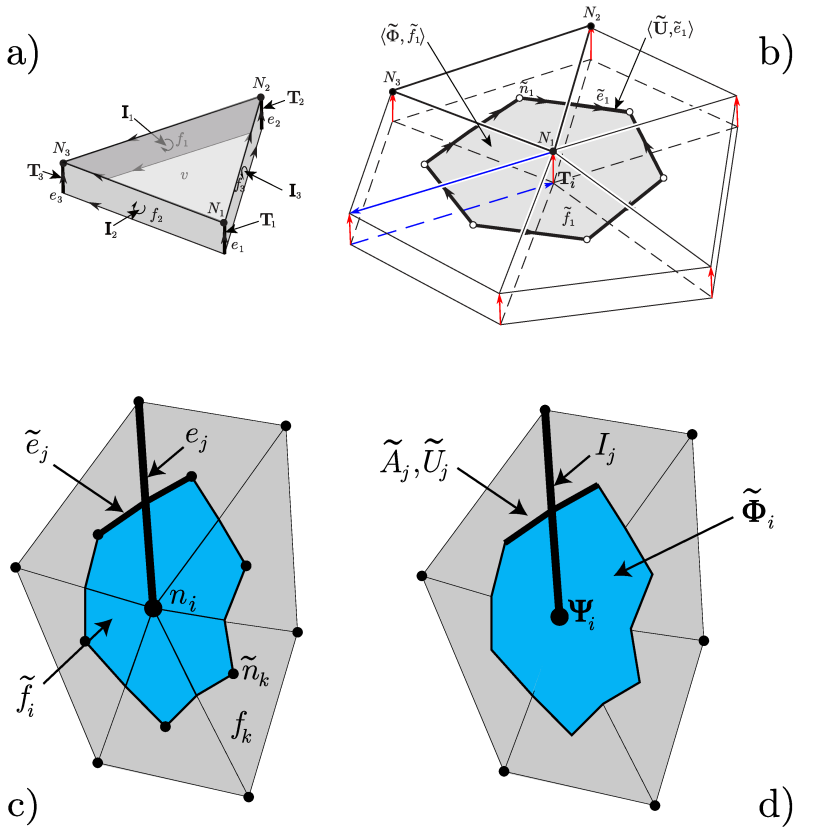

Figure 4.16: a) Geometric elements and primal physical variables of a 3D mesh $\mathcal{K}$ based on triangular prisms. b) The dual of $\mathcal{K}$ with the related global physical quantities associated to $\tilde{\mathcal{K}}$. In red, the *active* DOFs, in blue the annihilated ones. c) Reduction of the 3D mesh based on triangular prisms to a 2D simplicial mesh; primal and dual geometrical elements are shown. d) Association of the physical variables of the BI method to the geometric elements of the primal and dual 2D grids. The DOFs on the red edges are replaced with the nodal stream function $\mathbf{\Psi}$.

with the algorithm described in [65] which, again, is combinatorial and exhibits a linear time worst-case complexity. Finally, for the sake of precision, we also notice that in (4.33) the edges-nodes incidence matrix $\mathbf{G}$ has been used instead of $\mathbf{C}$ since, due to the considerations previously exposed, the current per unit of thickness is influenced only by the scalar values of T_i in correspondence of the 2D mesh nodes and this fact can be mathematically expressed by means of $\mathbf{G}$ and $\mathbf{\Psi}$ only.

Continuing the parallelism with the volume integral formulation, we recall again Faraday's discrete law referred to $\mathcal{K}$, that writes as

$$\mathbf{G}^T\tilde{\mathbf{U}} + i\omega\tilde{\mathbf{\Phi}} = -i\omega\mathbf{G}^T\tilde{\mathbf{A}}_s, \tag{4.34}$$

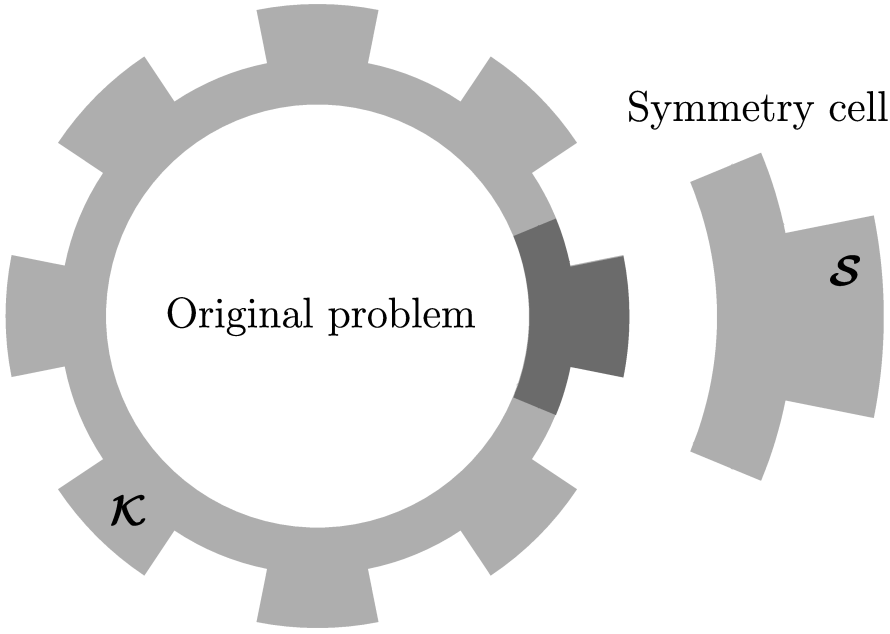

Figure 4.17: On the left: a thin conductor represented as a surface Σ encoded in the cell complex $\mathcal{K}$ exhibiting a cyclic symmetry whose group is of order eight. On the right: the related symmetry cell $\mathcal{S}$ representing 1/8 of the original problem.

in addition with the non-local law

$$\mathbf{H}^T\tilde{\mathbf{U}} + i\omega\mathbf{H}^T\tilde{\mathbf{A}} = -i\omega\mathbf{H}^T\tilde{\mathbf{A}}_s. \tag{4.35}$$

The symbols here used have the same meaning as in the previous sections but they are here referred to Σ as illustrated in figure 4.16d. Moreover, working on a two dimensional $\mathcal{K}$, $\tilde{\mathbf{C}} = \mathbf{G}^T$ holds.

Thanks to the already defined discrete constitutive law here recast to fit the 2D representation of the problem, $\tilde{\mathbf{U}} = \mathbf{RI}$ and $\tilde{\mathbf{A}} = \mathbf{MI}$, by substituting them and (4.33) into (4.34) and (4.35) the eddy current problems on Σ reads as

$$\begin{bmatrix} \mathbf{G}^T\mathbf{K}_\Sigma\mathbf{G} & \mathbf{G}^T\mathbf{K}_\Sigma\mathbf{H} \\ \mathbf{H}^T\mathbf{K}_\Sigma\mathbf{G} & \mathbf{H}^T\mathbf{K}_\Sigma\mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{\Psi} \\ \mathbf{i} \end{bmatrix} = \begin{bmatrix} -i\omega\mathbf{G}^T\tilde{\mathbf{A}}_s \\ -i\omega\mathbf{H}^T\tilde{\mathbf{A}}_s \end{bmatrix}. \tag{4.36}$$

where $\mathbf{K}_\Sigma := \mathbf{R} + i\omega\mathbf{M}$.

4.3.2 How to exploit cyclic symmetry

As announced, a wise way to reduce the size of the dense matrix (4.36) representing the eddy current problem with a BI formulation is exploiting *cyclic symmetry* when possible. This reflects on solving a family of equivalent problems on the *symmetry cell* $\mathcal{S}$ only instead of dealing with its solution on the whole cell complex $\mathcal{K}$ (refer to figure

4.17 for a visual interpretation of $\mathcal{K}$ and $\mathcal{S}$). To that end, the most common approach to reduce the problem from $\mathcal{K}$ to $\mathcal{S}$ makes use of *Discrete Fourier Transform* (DFT) as proposed, for instance, in [63] to decompose the full problem into a series of subproblems on $\mathcal{S}$.

A fundamental requirement to apply DFT in case of integral formulations is that the system matrix of (4.36) has to be symmetric and in the form

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_1 & \mathbf{S}_2 & \mathbf{S}_3 & \cdots & \mathbf{S}_{S_g-1} & \mathbf{S}_{S_g} \\ \mathbf{S}_{S_g} & \mathbf{S}_1 & \mathbf{S}_2 & \cdots & \mathbf{S}_{S_g-2} & \mathbf{S}_{S_g-1} \\ \mathbf{S}_{S_g-1} & \mathbf{S}_{S_g} & \mathbf{S}_1 & \cdots & \mathbf{S}_{S_g-3} & \mathbf{S}_{S_g-2} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{S}_3 & \mathbf{S}_4 & \cdots & \cdots & \mathbf{S}_1 & \mathbf{S}_2 \\ \mathbf{S}_2 & \mathbf{S}_3 & \cdots & \cdots & \mathbf{S}_{S_g} & \mathbf{S}_1 \end{pmatrix}, \tag{4.37}$$

in which S_g stands for the order of the *cyclic group* whose definition can be found in [69] ($S_g = 8$ for the problem in figure 4.17) and where necessarily $\mathbf{S}_i = \mathbf{S}_{S_g-i+2}^T,\ \forall i > 1$.

The block matrix $\mathbf{S}_1$ represents the entries of the left hand side of (4.36) written just on the symmetry cell $\mathcal{S}$, $\mathbf{S}_2$ stems from (4.36) referred to the inductive mutual terms between two distinct symmetry group an so. Such a matrix is called *block-circulant*.

As first important remark, it is here claimed that this matrix when written for the BI formulation is not guaranteed to be block-circulant, generally. In fact, if we consider the non-trivial domain $\mathcal{K}$ of figure 4.18 and we construct a representative of the cohomology generator by using the algorithm in [65] it may happen that the obtained generator *spoils* the symmetry of the problem as in the case of the generator in figure 4.18. As one may hope, the solution of equation (4.36) does not depend on the choice of the generator given that the basis represented is the same and thus, no requirements are necessary on the generator shape when solving the whole problem on $\mathcal{K}$. Differently, if the solution has to be computed on the symmetry cell $\mathcal{S}$ only also the generator has to share the same cyclic symmetry in order to make the matrix block-circulant. As a drawback, this implies that a generator computed on $\mathcal{K}$ cannot be reused and adapted on $\mathcal{S}$.

As a consequence, a recipe to assure the cyclic symmetry of the generators has to be provided in order to preserve the block-circulant property of the system matrix. Once that this property is satisfied, the approach whose details are described in [64], can be applied to reduce the problem size and solve one or more problems on $\mathcal{S}$. If the source of magnetic field is symmetric too, the DFT application yields a unique problem to be solved whose system matrix $\mathbf{Q}$ (i.e. a proper right hand side of the system written on $\mathcal{S}$ which has to be equivalent to (4.36)) is computed as

$$\mathbf{Q} = \sum_{j=1}^{S_g} \mathbf{S}_j. \tag{4.38}$$

where $\mathbf{S}_j$ is the jth block of (4.37).

How to compute the basis of the representatives of cohomology generators to assure the matrix to be block-circulant?

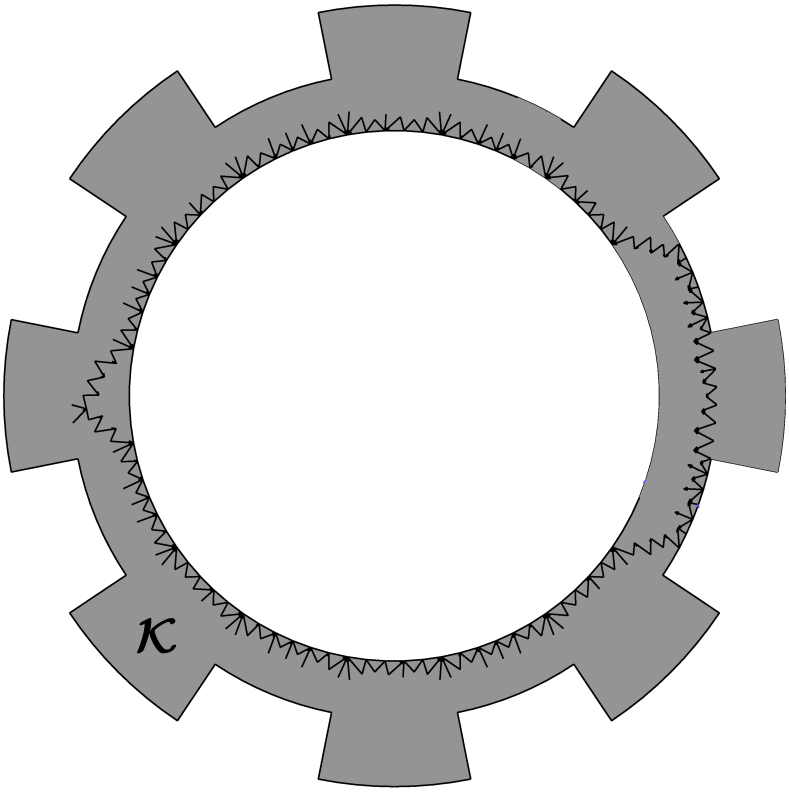

Figure 4.18: The complete geometry encoded into $\mathcal{K}$ where a cohomology generator (black edges) has been computed with a "standard" algorithm.

4.3.3 Cyclic symmetry and cohomology computation on a 2D manifold

In the former section, it was expressed the idea that the retrieval of the cohomology generator affects the properties of the system matrix thus allowing or not the exploitation of the cyclic symmetry of the problem. Specifically, the generators to be built should fulfil this two requirements:

- the computation should be performed on the symmetry cell $\mathcal{S}$ to avoid the construction of the whole geometry $\mathcal{K}$ thus saving time when meshing the surface Σ representing the problem. In regard to this point, it is here remarked that since the computation of the cohomology basis takes just few seconds because of the combinatorial characteristic of the algorithm, one should compute it on the whole $\mathcal{K}$ since this would not considerably affect the overall simulation time. On the contrary, this requires the construction and the meshing of the whole $\mathcal{K}$ that is a more time consuming task, instead;

- the retrieved generators must share the same cyclic symmetry of the whole geometry $\mathcal{K}$.

Since the combinatorial algorithm for the cohomology generators retrieval proposed in [65] can act on an arbitrary orientable surface, there is no reason for which one should not apply it to the cell complex $\mathcal{S}$. Unfortunately, just retrieving the $H^1(\mathcal{S}-\partial\mathcal{S}) \simeq H^1(\mathcal{S},\partial\mathcal{S})$ cohomology generators on $\mathcal{S}$ does not lead to any result since generally the symmetry cell does not preserve the same topological properties of $\mathcal{K}$. In fact, in the example proposed in figure 4.17, it is plain to see that $\mathcal{S}$ is trivial whereas $\mathcal{K}$ it is not.

In order to work on $\mathcal{S}$ only and recover the topological properties of $\mathcal{K}$ two additional features has to be added to the cohomology algorithm in [65]:

a) to perform the computation on S, instead of a basis on $H^1(\mathcal{S},\partial\mathcal{S})$ a basis that belongs to the first relative cohomology group $H^1(\mathcal{S},\partial\mathcal{S}-(B_1\cup B_2))$ has to be used. The boundaries of the symmetry cell here denoted B_1, B_2 are shown in figure 4.19 and they are defined as that edges that belongs to $\mathcal{S}\cap\partial cK$;

b) since the requirement for the generators to belong to $H^1(\mathcal{S},\partial\mathcal{S}-(B_1\cup B_2))$ does not re-establish the topological properties of $\mathcal{K}$ on $\mathcal{S}$ we make also use of the idea represented in figure 4.20: to build representatives with the same symmetry of $\mathcal{K}$, we stick together the symmetric boundaries B_1 and B_2 identifying them as a unique topological subset of $\mathcal{S}$, then the modified algorithm in [65] to retrieve generators on $H^1(\mathcal{S},\partial\mathcal{S}-(B_1\cup B_2))$ is applied to the so obtained new cell complex. By this way, the cyclic symmetry properties of $\mathcal{K}$ are assured also on $\mathcal{S}$.

As a result, in figure 4.19 the achieved cohomology generator that shares the same cyclic symmetry of $\mathcal{K}$ is shown. We also state that, for the procedure to be successful, the only hypothesis we assume on the symmetry cell boundaries $\mathcal{S}\cap\partial cK$ is that the mesh is built in such a way that B_2 can be obtained by rotating $B1$ in order to create a one-to-one correspondence between the edges to be glued together and preserve the cyclic symmetry. This assumption holds in case of a hole crossing the cutting plane of the symmetry too.

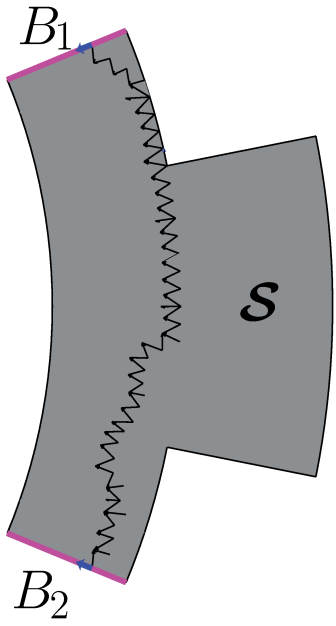

Figure 4.19: A generator on $\mathcal{S}$ that belongs to the relative first cohomology group $H^1(\mathcal{S}, \partial\mathcal{S} - (B_1 \cup B_2))$. The symmetry boundaries B_1 and B_2 are highlighted in violet; two topologically coincident edges that belongs to the support of the generator on B_1 and B_2 are drawn in blue.

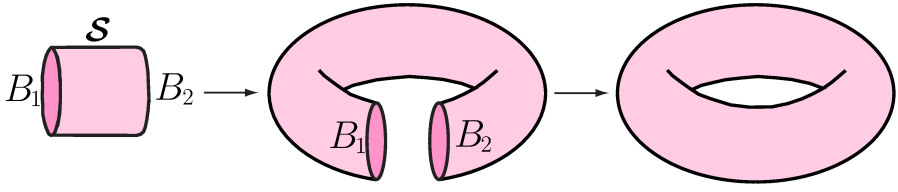

Figure 4.20: Topological stitching on the symmetry cell $\mathcal{S}$: the two boundaries B_1, B_2 are topologically glues together to recover a non-trivial domain as it was for $\mathcal{K}$.

4.3.4 Numerical results

To validate the method here proposed we compare the solution obtained on the whole thin conductor $\mathcal{K}$ represented in figure 4.17 to that one obtained on its symmetry cell $\mathcal{S}$ to verify the correspondence between the two solutions. Two test cases were considered in order to study the computation speed up with both a symmetric source and a non-symmetric one.

The result related to the axisymmetric source is shown in figure 4.21: the source is represented as a red circular wire in which a current $I_c = 1\,\mathrm{A}$ flows at a frequency $f = 3.5\,\mathrm{MHz}$. The two mesh of $\mathcal{K}$ and $\mathcal{S}$ have 9881 and 1247 DOFs, respectively. The related speed up evaluated in terms of wall time to get the solution is about 23 which is much higher than the symmetry group ratio[7] thanks to the additional saving during the matrix assembly: due to the symmetry of the domain, the geometrical entities required during the computation of $\mathbf{M}$ for each single block matrix $\mathbf{S}_j$ (refer to eq. (4.38)) can be computed on the fly by rotating the symmetry cell $\mathcal{S}$ thus saving the time related to the memory access when retrieving the mesh entities (triangles, nodes, incidence matrices). In fact, since to obtain the entries of $\mathbf{M}$, nodes, primal and dual edges, for each couple of volumes $v_i \in \mathbf{S}_1$ and $v_j \in \mathbf{S}_2$ are required, once these geometrical entities are obtained for v_j we can easily compute also the corresponding nodes, primal and dual edges on the corresponding rotated volumes $v_k \in \mathbf{S}_{2+w}, w = 1, ..., S_g - 2$ thus populating also the entries of the $\mathbf{S}_{2+w}, w = 1, ..., S_g - 2$ blocks without interrogating the mesh at each step.

In relation to this test, it has also been verified that if the mesh of $\mathcal{K}$ is obtained by repeating the mesh of the symmetry cell, then the two solutions on the whole conductor and on the symmetry cell are the same, up to linear solver tolerance.

As a second step, we tested the implementation in the case of a general non-symmetric source too, whose result for the complete geometry can be found in figure 4.22. In this case the resulting speed up is about 16: this is due to the fact that in case of a non-symmetric source the DFT application produces a family of subproblems to be solved on $\mathcal{S}$ in order to reconstruct the solution on $\mathcal{K}$. Hence, the assembly of the system matrix results to be less efficient thus increasing the computational time.

It is thus natural to notice that the exploitation of cyclic symmetry in case of an axisymmetric source is much more appealing than the case in which an arbitrary source has to be studied. In this last case, more than one equations system has to be assembled and solved thus reducing the advantage with respect to the problem solution on $\mathcal{K}$.

4.3.5 Cyclic symmetry for a volume integral formulation

We now extend the main ideas introduced for the boundary integral method to the volume integral code for polyhedral grids. With respect to the prior situation we have mainly two novelties: first, in dealing with polyhedra, a novel algorithm to compute the cohomology generators on non-trivial domains is proposed; this new algorithm aim at reusing any code for triangulated surfaces recasting it for a 2D mesh built with polygons in order to reduce the implementation effort. Second, as already forewarned in section 4.2.1, when working on the symmetry cell instead of the whole 3D geometry, also the technique to gauge the DOFs of the problem has to be tweaked accordingly with

[7] Theoretically, since in this case $S_g = 8$, only 1/8 of the matrix has been assembled thus one would expect at least a speed up equal to 8.

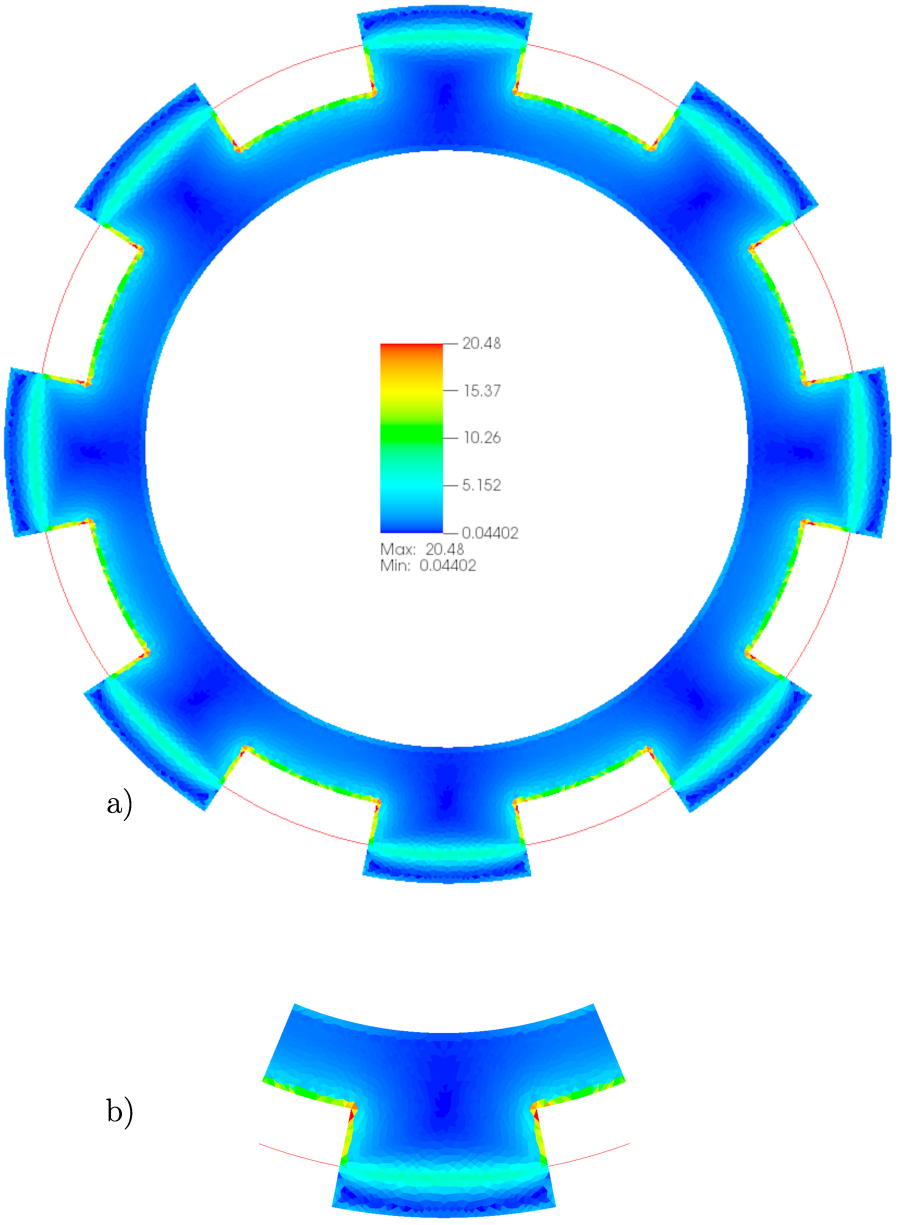

Figure 4.21: The solution a) on the whole mesh $\mathcal{K}$ and b) on the symmetry cell $\mathcal{S}$ in case of an axisymmetric source (in red).

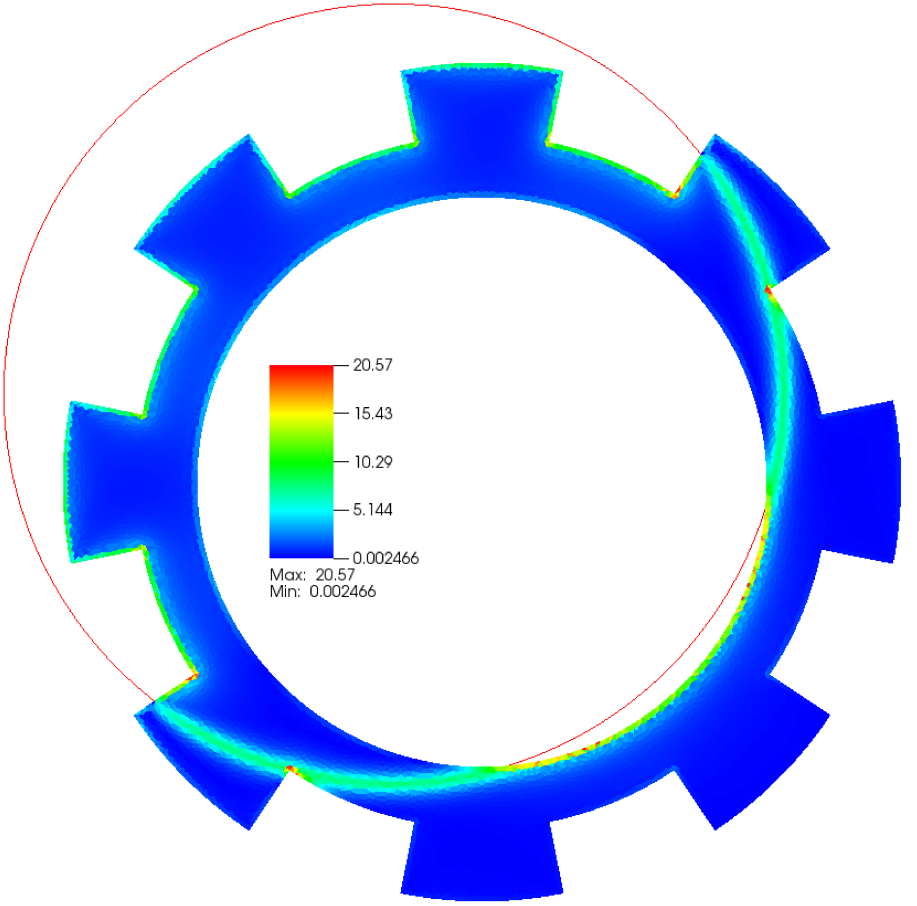

Figure 4.22: Reconstruction of the real part of the current density distribution on the whole mesh $\mathcal{K}$ from the solution of the subproblems on $\mathcal{S}$, in case of a general source here represented by the red wire.

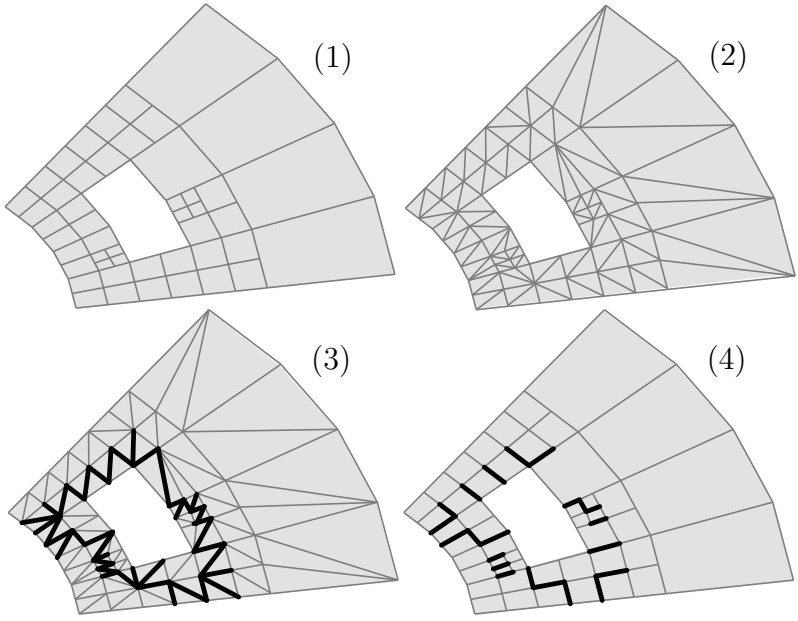

Figure 4.23: A part of the polygonal mesh of $\partial\mathcal{K}$. The four steps of the algorithm are described: 1) the polygonal mesh; 2) a triangulation of the polygonal mesh; 3) the cohomology generator computed with a software for triangular grids; 4) the cohomology generator on the polygonal mesh.

the new topological setting produced by the reduction of the computational domain to a symmetry cell. In fact, in a similar way to what happens for the cohomology computation, also a wrong construction of the tree-cotree decomposition can lead to a non-block-circulant matrix because of the spoiling of the cyclic symmetry. For the sake of precision, we here remember that exploiting the symmetry of the problem allows a reduction of the DOFs and thus of the dense matrix size; notwithstanding, matrix compression techniques are absolutely not discarded as equally effective ways to achieve the same result and indeed they can be used in addition to the proposed approach to further reduce the system matrix size.

4.3.6 Cohomology computation on a polygonal surface

In literature, various softwares are provided for the cohomology computation on a 2D simplicial cell complex such as [54, 70, 71, 72]. Differently, despite some approaches applicable to polyhedral grids have been already proposed in literature as in [46] or [59], in the author's opinion, a robust implementation in a off-the-shelf software or library able to deal with such polygonal grids is not trivial thus such a tool is still far to be fully general. For instance, to achieve effectiveness and robustness in the retrieval of the generators for the VI code we resorted to lazy generators that, however, provide twice the number of the required cycles and for this reason they can be applied just for some problems as the ones in the frequency domain we have considered before to validate the

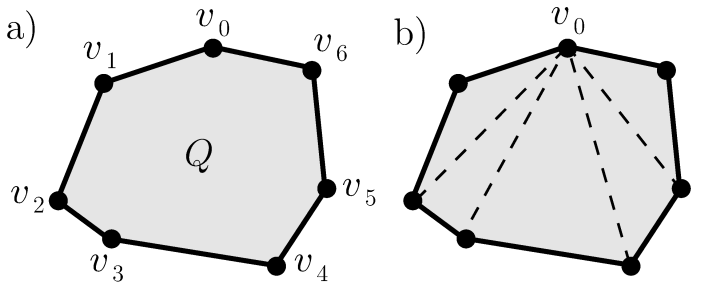

Figure 4.24: a) A polygon Q; b) topological triangulation of Q: no new nodes are added.

code. For this reason, we here suggest an alternative method to reuse more robust and general software for triangulated 2D meshes.

Following the diagrams in figure 4.23, the main steps to apply a software for triangulations to a polygonal surface are:

1. starting from the 3D computational domain Ω_c encoded into the cell complex $\mathcal{K}$, the boundary $\partial\mathcal{K}$ is considered first. To better understand the successive steps only a part of this boundary $\partial\mathcal{K}$ is represented in figure 4.23, step 1;

2. a partition of the 2D polygonal mesh is performed in order to obtain a triangulation of $\partial\mathcal{K}$, as shown in figure 4.23, step 2. Given that the main constraint of this procedure is that no new node can be inserted, this step can be easily implemented by means of the incidence matrices $\mathbf{C}$ and $\mathbf{G}$ reduced to the boundary faces of $\mathcal{K}$ (previously named $\mathbf{C}_b$, $\mathbf{G}_b$, respectively) by adding the remaining topological connections (i.e. "virtual" edges) for each set of nodes belonging to a polygon. In fact, if Q is a polygon identified by the vertices v_0, v_1, v_2, ..., v_n in this precise order (see figure 4.24a) and a chord in Q is a line segment that connects two nonadjacent vertices in P, then if $n-3$ chords are drawn in Q as a consequence $(n-2)$ triangles are obtained that *triangulate* the polygon Q as depicted in figure 4.24b. Precisely, the triangles are identified by the list of nodes $\{v_0, v_1, v_2\}$, $\{v_0, v_2, v_3\}$, $\{v_0, v_3, v_4\}$, ..., $\{v_0, v_{n-1}, v_n\}$. As additional remark, we here recall that the partitioning is just topological, thus no new edges are effectively added to the cell complex $\mathcal{K}$;

3. once the triangulation is available, the $H^1(\partial\mathcal{K})$ cohomology generators are retrieved on the 2D simplicial mesh by means of a software for triangulated surfaces. In figure 4.23 (step 3) an example of the support of the representative of the cohomology generator for the triangulated $\partial\mathcal{K}$ is reported;

4. as last step, the matrix $\mathbf{H}$ storing the basis of the cohomology generators is built from the coefficients of the generators computed on the triangulated mesh. Each edge belonging to the generator of the simplicial mesh is represented as a pair of nodes: if that pair of nodes identifies an edge in the $\mathbf{G}_b$ matrix of the original polygonal mesh thus belonging to the original $\mathcal{K}$ too, then the edge is assembled into $\mathbf{H}$ in its correspondent position; if not, the edge is identified as one of the additional topological edges and it is thus discarded.

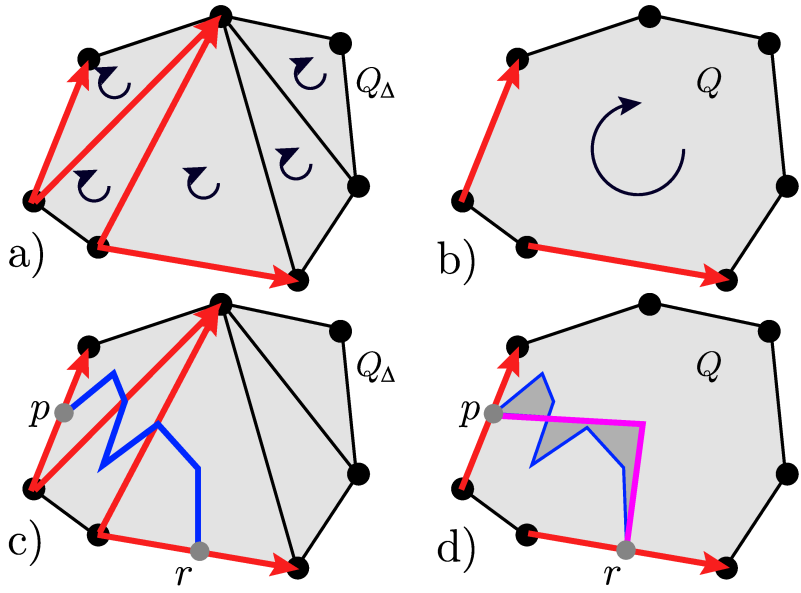

Figure 4.25: a) A 1-cochain $\mathbf{c}_\triangle$ on $Q_\triangle$ (red thick edges) that is a 1-cocycle since their circulation is zero due to the triangles orientation (circular black arrows). b) the cocycle $\mathbf{c}_\triangle$ of $Q_\triangle$ is still a cocycle on Q too because Q is a sum of triangles of $Q_\triangle$. c) The dual cycle (blue edges) which is dual to $\mathbf{c}_\triangle$ in $Q_\triangle$. d) A part of the dual cycle (magenta edges) which is dual to $\mathbf{c}'$ limited to Q . Both this dual cycle and the one in figure 4.25c start in p and end in r. The dark grey area is their difference, thus they are homologous.

The remaining step is now proving that if $\mathbf{c}$ is the array of coefficients expressing the cohomology generator found in the triangulated mesh of $\partial\mathcal{K}$ then $\mathbf{c}$ produces a representative for the polyhedral mesh too in the aforementioned way. This can be achieved by first demonstrating that the circulation of the 1-cochain $\mathbf{c}$ on $\mathcal{K}$ is zero i.e. $\mathbf{c}$ is a 1-cocycle on $\mathcal{K}$ and then show that the dual cycle expressed by $\mathbf{c}$ is homologous to the dual cycle obtained by $\mathbf{c}'$ which is $\mathbf{c}$ reported on the original $\partial\mathcal{K}$.

As far as the first goal is concerned, by taking advantage of the illustration in figure 4.25a-b, we can focus on a single polygon Q (since $\partial\mathcal{K}$ can be obtained as a linear combination of polygons $Q_i,\ i = 1, ..., F$) and notice that Q can be in turn expressed as a linear combination of the triangles forming its triangulation $Q_\triangle$. More than this, the software for the cohomology computation on $Q_\triangle$ fulfil the requirement

$$\mathbf{C}_\triangle \mathbf{c}_\triangle = \mathbf{0}$$

where $\mathbf{C}_\triangle$ is the faces-edges incidence matrix on $Q_\triangle$ and $\mathbf{c}_\triangle$ is the 1-cocycle $\mathbf{c}$ restricted to $Q_\triangle$. Since this means that the sum of the coefficients of the circulation of $\mathbf{c}_\triangle$ is zero on $Q_\triangle$ then from this two considerations it follows that the circulation is zero also on Q because Q is a linear combination of $Q_\triangle$. This proves that $\mathbf{c}'$ in $\partial\mathcal{K}$ is a 1-cocycle too.

Second, in regard to the homology between the dual cycle of $\mathbf{c}'$ restricted to Q and

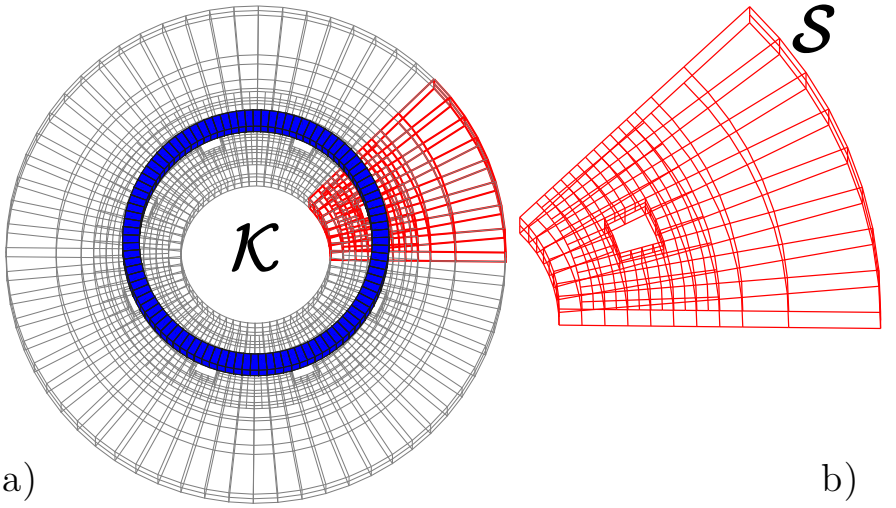

Figure 4.26: a) The complete 3D geometry (disk with 9 holes) exhibiting cyclic symmetry on which subgridding has been applied by means of polyhedral elements; the geometry is encoded into the cell complex $\mathcal{K}$. b) The symmetry cell encoded into the cell complex $\mathcal{S}$.

the dual cycle $\mathbf{c}_\triangle$ in $Q_\triangle$, we can say that the property is verified since these two dual 1-cocycles differs for a boundary only, as explained in figure 4.25c-d.

4.3.7 Cyclic symmetry and gauging for the VI code

Let us now explain how to deal with cyclic symmetry when solving eddy currents by means of the VI code. Similarly to what has been done for the BI method, also in this case we resort to the Discrete Fourier Transform in order to solve an equivalent family of one or more sub-problems on the symmetry cell $\mathcal{S}$ instead of solving the whole problem on $\mathcal{K}$. The whole geometry and the symmetry cell which we refer to are reported in figure 4.26.

As announced, when working on a 3D geometry there are two aspects to be taken into account because they can spoil the symmetry of the problem thus leading to have a non-block-circulant matrix. The first issue is related to cohomology generators and, indeed, it can be fixed with the same approach applied to the BI method. As an example, if we consider figure 4.27a, we notice that the retrieved generator cannot be used to solve the problem on $\mathcal{S}$. Moreover, also in this case, it would be useful to perform all the computation by considering the symmetry cell only thus avoiding the construction of the cell complex $\mathcal{K}$. Hence, if we consider $\mathcal{S}$, by applying the same idea represented in figure 4.20 i.e. by glueing the two boundaries B_1 and B_2 of the 3D symmetry cell, we can obtain the required $H^1(\partial\mathcal{S}, B_1 \cup B_2)$ generators thanks to the algorithm exposed in the previous section. The support of such generators is represented in figure 4.27b.

We remark that after the cohomology computation has been performed on the *glued*

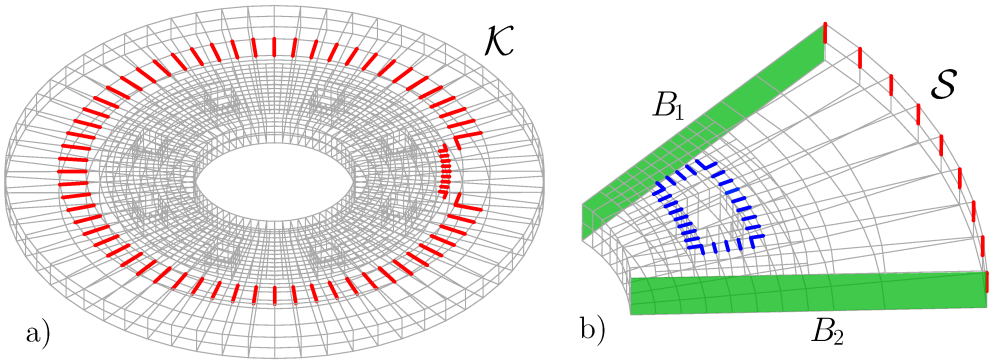

Figure 4.27: a) The support of a cohomology generator for the whole geometry $\mathcal{K}$. b) The supports of the two generators for the symmetry cell $\mathcal{S}$; the symmetry boundaries of $\mathcal{S}$ are represented in green.

cell complex where nodes, edges and faces of B_1 and B_2 are encoded as the same topological entities, the generator is "rewritten" on the two original distinct boundaries by copying the coefficients in the two distinct entries of the matrix $\mathbf{H}$ (referred to the distinct edges of B_1 and B_2). In addition, taking advantage of figure 4.27b, we may notice that *local* generators as the one depicted in blue, never spoil the block-circulant property of the problem because they refers to edges of $\mathcal{S}$ complex only and so they can be stored into the problem matrix as local entries of a specific block S_j; differently, *global* generators as the red one in the figure, could lead to a not block-circulant matrix if they do not share the same symmetry of $\mathcal{K}$. Indeed, if the global generator is symmetric which means that is repeated *exactly* in each symmetry cell of $\mathcal{K}$ this means that every part of the generator crossing a symmetry cell is crossed by the same nonlocal current: this allows to ideally rearrange the non-local terms as local entries of each S_j block. Conversely, this reasoning is not possible thus explaining the need of symmetric generators both for the BI and VI formulation when reducing the problem from $\mathcal{K}$ to $\mathcal{S}$.

As second novel aspect to be treated, let us discuss about the construction of a proper tree-cotree decomposition to gauge the solution and obtain a full rank system with the minimum number of DOFs. By recalling the information exposed in section 4.2.1, we stated that there are two possible equivalent ways to build the tree-cotree decomposition. Yet, we here explain that one of the two ways is not suitable to deal with cyclic symmetry because the obtained tree if built on the whole geometry $\mathcal{K}$ spoils the symmetry whereas if built on the symmetry cell $\mathcal{S}$ it does not produce a block circulant matrix because it is not a tree for $\mathcal{K}$ anymore.

In detail, if we focus on figure 4.28a where the tree-cotree decomposition is performed by computing two independent trees on $\partial\mathcal{K} \cap \mathcal{S}$ (violet dotted edges) and in the inner part of $\mathcal{S}$ (red thick lines) then joined one to the other (via the green edge in the picture), we may notice that when this decomposition is considered in the whole conductor $\mathcal{K}$ this tree forms cycles on $\mathcal{K}$ thus resulting a non considerable tree. On the other hand, if the spanning tree is built by expanding a tree from the boundary of the symmetry cell towards its inner part then the symmetry is respected and the tree does not form cycles

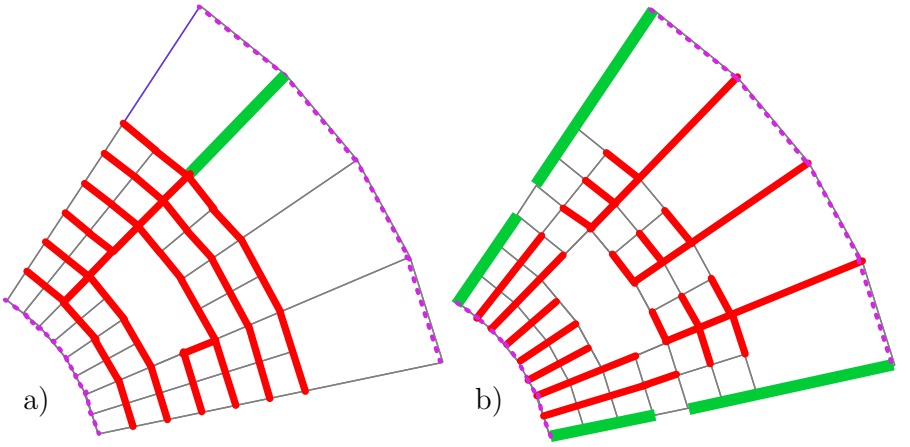

Figure 4.28: Comparison between the two possible approaches for the spanning tree construction (upper view of $\mathcal{S}$). a) the tree on $\mathcal{S}$ forms cycles on the complete cell complex $\mathcal{K}$; b) the symmetric tree-cotree decomposition on $\mathcal{S}$ does not form cycles if extended to $\mathcal{K}$. In both cases, the tree built on $\mathcal{S} \cap \partial\mathcal{K}$ represented by the violet dotted edges, is not subjected to any requirement since all the DOFs of the symmetry cell boundary are set to zero because of the BCs applied.

on $\mathcal{K}$. More precisely: if $\mathcal{S}'$ is the symmetry cell where the topological stitching of B_1 and B_2 has been applied, we first construct a spanning tree on $\partial\mathcal{S}'$ and then we expand the obtained tree into $\mathcal{S}' \setminus \partial\mathcal{S}$. By this way, the resultant tree-cotree decomposition does not form cycles in $\mathcal{K}$ because it would form a cycle on $\mathcal{S}'$ too. We also recall that the preliminary tree on $\partial\mathcal{S}'$ (the violet edges in the figure) is not subjected to any requirements in terms of symmetry since all the DOFs related to any surface edge are annihilated because of the boundary conditions.

This approach is exemplified in figure 4.28b where the tree on the boundary $\partial\mathcal{S}'$ is represented with the violet dotted lines whereas the expanded tree in $\mathcal{S}' \setminus \partial\mathcal{S}'$ is denoted with the red thick edges: in green the edges of the topologically coincident boundaries B_1, B_2 that belong to the resultant tree-cotree decomposition. Once the tree on $\mathcal{S}'$ is found, the edges of the tree belonging to the glued $B_1 \equiv B_2$ are encoded into $\mathcal{S}$ by copying the coefficients into the distinct edges of B_1 and B_2.

4.3.8 Numerical results

To benchmark the proposed approach, eddy current problem was solved on both the geometries represented in figure 4.26. The axisymmetric source of magnetic field employed is a stranded coil (depicted in blue in the figure) in which a current $I_c = 100\,\mathrm{At}$ at a frequency $f = 200\,\mathrm{Hz}$ flows. The result in terms of real current density $re\{\mathbf{J}\}$ field is shown in figure 4.29: the current density field calculated on the whole 3D geometry $\mathcal{K}$ is the same as the one on the symmetry cell $\mathcal{S}$ up to the tolerance of the linear solver.

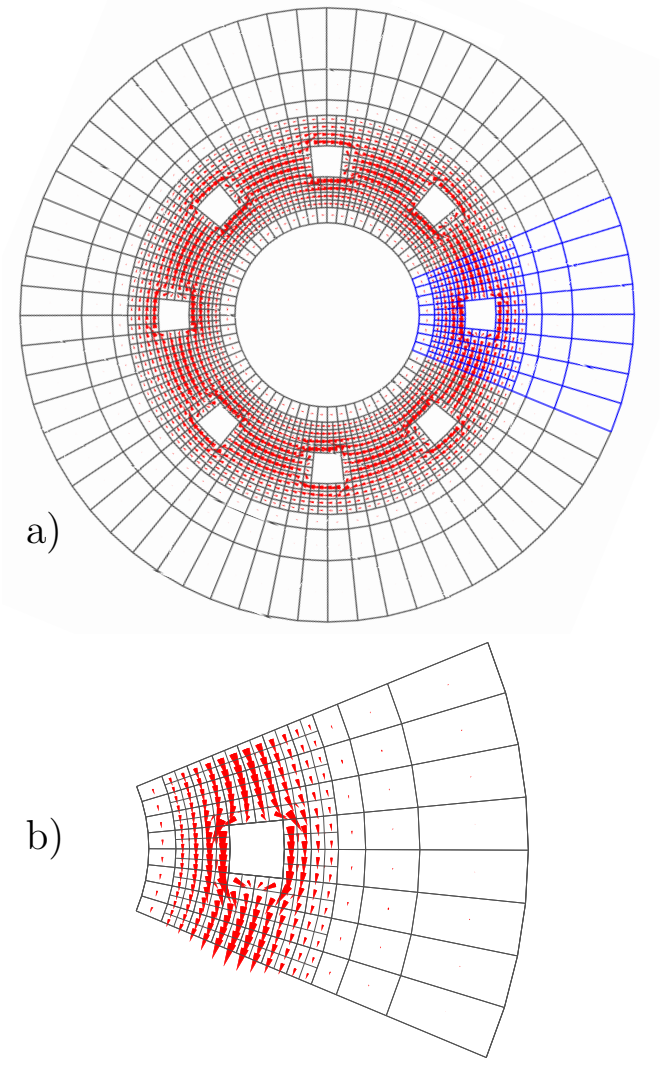

Figure 4.29: a) Real part of the current density $re\{\mathbf{J}\}$ in the solid conductor $\mathcal{K}$ b) $re\{\mathbf{J}\}$ in $\mathcal{S}$.

5

Iterative schemes to solve eddy currents with integral formulations

Besides exploiting the symmetry of the conductor in which eddy currents flows, another effective technique to reduce the size of the problem matrix to be assembled is represented by iterative formulations.

Originally, iterative techniques have been employed for the solution of electromagnetic problems in order to find a sound alternative to the use of direct solvers that in the past were way less fast and efficient than nowadays. For this reason, especially when the goal was the computation of eddy currents by means of integral formulations, some authors started studying the effectiveness of this method applied to T-formulation [73, 74]. In that case, the entries of the problem matrix were always computed and then assembled in such a way to lead to a Jacobi iteration scheme. Then, the key point to be studied became the convergence of the scheme which, as we are going to see, it was and it is still affected by the physical parameters of the problem as the resistivity, the frequency and the domain size [75].

Starting from this premise, we here propose a method to take advantage of iterative schemes in order *to reduce the number of the entries* that has to be assembled and then stored: the root idea of this approach stems from rewriting the system of equation in such a way that only a part of the complete system in (4.30) or (4.36) is assembled whereas the "missing" entries are taken into account in an alternative way. Mainly, we exploit the post processing tools to efficiently compute Biot–Savart law on the fly instead of discretizing the related integral constitutive law. Nowadays, this can be efficiently done since the computation is parallelizable and it can be performed on GPUs, as shown in [76]. Yet, in regard to convergence trends, some non-negligible limits arise also in this case.

For this reason, as further step, when Ω_c is constituted by many distinct conductors so that $\Omega_c = \Omega_1 \cup \Omega_2 \cup ... \cup \Omega_N$ and $\Omega_i \cap \Omega_j = \emptyset$, $\forall i \neq j$, we will recast the previ-

ous iterative scheme in order to improve the convergence behaviour. This is achieved through the exploitation of two additional features: on one hand, we take advantage of domain decomposition techniques applied to the distinct Ω_i conductors, each of them regarded as a sub-domain; on the other hand, Krylov subspace techniques [77] are used to improve the convergence trend when the standard Gauss–Seidel scheme requires too many iterations in order to provide an accurate solution. Moreover, we claim that this iterative method is not alternative to the use of matrix compression techniques that can be additionally used to further reduce the size of the fully populated matrices obtained for each sub-domain.

As final remark, we specify that in this chapter we will focus our attention on the convergence of the proposed schemes only since this topic is fundamental for the methods to be applied. As a consequence, the aspects related to a parallel and robust implementation of Biot–Savart law computation are not addressed since they are already delineated in [76].

5.1 A volume integral iterative formulation

The first investigated approach stems from the volume integral formulation of chapter 4. Hence, the iterative scheme that we are going to develop, aims at solving eddy currents on a polyhedral mesh of a solid conductor of arbitrary topology.

For the sake of simplicity, we temporarily assume that the computational domain is topologically trivial. Thus, since in this case $\mathbf{H} = \mathbf{0}$, the system of equations in (4.30) reduces to

$$(\mathbf{K}_R + i\omega\,\mathbf{K}_M)\mathbf{T} = \mathbf{b}_s, \tag{5.1}$$

where we have defined

$$\mathbf{K}_R = \mathbf{C}^T\mathbf{R}\mathbf{C}$$

and

$$\mathbf{K}_M = \mathbf{C}^T\mathbf{M}\mathbf{C},$$

and where we recall that

$$\mathbf{b}_s = -i\omega\mathbf{C}^T\tilde{\mathbf{A}}_s.$$

Since, as explained in section 4.1, $\mathbf{K}_M$ is a fully populated matrix whereas $\mathbf{K}_R$ is very sparse, we aim at separating this two contributions similarly to what is proposed in [73, 74]. Thus, equation (5.1) becomes

$$\mathbf{K}_R\,\mathbf{T} = -i\omega\,\mathbf{K}_M\,\mathbf{T} + \mathbf{b}_s, \tag{5.2}$$

from which it is possible to obtain $\mathbf{T}$ as

$$\mathbf{T} = -i\omega\,\mathbf{K}_R^{-1}\,\mathbf{K}_M\,\mathbf{T} + \mathbf{K}_R^{-1}\,\mathbf{b}_s. \tag{5.3}$$

This last equation is in a form suitable for the application of a *fixed point* or *Jacobi* iterative scheme, that can be expressed as

$$\mathbf{T}^n = -i\omega\,\mathbf{K}_R^{-1}\,\mathbf{K}_M\,\mathbf{T}^{n-1} + \mathbf{K}_R^{-1}\,\mathbf{b}_s. \tag{5.4}$$

Indeed, this scheme clearly does not yield any benefit since we still have to build $\mathbf{K}_M$ matrix and store it into the calculator memory. For this reason, we recast equation (5.4) thanks to (4.26) and (4.27b) which leads to

$$\mathbf{K}_M \mathbf{T} = \tilde{\mathbf{\Phi}} = \mathbf{C}^T \tilde{\mathbf{A}}, \tag{5.5}$$

in which each entry $\tilde{A}_i,\ i = 1, ..., F$ of $\tilde{\mathbf{A}}$ has been defined as the integral of the magnetic vector potential $\boldsymbol{a}(\boldsymbol{r})$ along the dual grid edges

$$\tilde{A}_i = \int_{\tilde{f}_i} \boldsymbol{a}(\boldsymbol{r}) \cdot \hat{\boldsymbol{t}}\, dl.$$

This allows to express $\mathbf{K}_M$ in an alternative way by directly computing the unknown $\boldsymbol{a}(\boldsymbol{r})$ starting from a guessed $\mathbf{T}$ value at each step of the iteration scheme. Hence, we resort to the expression for the magnetic vector potential calculation of a polyhedral source of current contained in [61] which, as said, is accurate and parallelizable, to compute $\boldsymbol{a}(\boldsymbol{r})$.

In the sequel, we provide further details on the developed iterative algorithm and then we also extend the approach to deal with non-trivial conductors by applying cohomology computation in the form of lazy generators, specifically.

5.1.1 The iterative scheme

If equation (5.5) is substituted into equation (5.4) the following iterative scheme, valid for a simply connected domain, is achieved:

$$\mathbf{K}_R \mathbf{T}^n = -i\omega \tilde{\mathbf{\Phi}}_c^{n-1} + \mathbf{b}_s \tag{5.6}$$

where $\tilde{\mathbf{\Phi}}^{n-1} = \mathbf{C}^T \tilde{\mathbf{A}}^{n-1}$ and with $\tilde{\mathbf{A}}^{n-1}$ that is directly computed using Biot–Savart law from the values of current density in a volume v of the mesh at the $(n-1)$ step as

$$J_v^{n-1} = \frac{1}{|v|} \sum_{j=1}^{F_v} \tilde{f}_j (\mathbf{C}\mathbf{T}^{n-1})_j. \tag{5.7}$$

The symbol $(\mathbf{C}\mathbf{T}^{n-1})_j$ represents the jth entry I_j of the current array $\mathbf{I}$ obtainable by starting from the DOFs array at the $(n-1)$ step $\mathbf{T}^{n-1}$, namely $\mathbf{I}^{n-1} = \mathbf{C}\mathbf{T}^{n-1}$.

The whole iterative procedure above described is represented in the scheme of figure 5.1. The algorithm is launched by setting $\mathbf{T}^0 = \mathbf{0}$ to then obtain the first value of the DOFs array

$$\mathbf{T}^1 = -i\omega \mathbf{K}_R^{-1} \tilde{\mathbf{\Phi}}^0 + \mathbf{b}_s$$

from which the iteration successively is kept running until the current array update

$$\frac{|\mathbf{I}^n - \mathbf{I}^{n-1}|}{\alpha \cdot |\mathbf{I}^1|} = \frac{|\mathbf{C}^T\mathbf{T}^n - \mathbf{C}^T\mathbf{T}^{n-1}|}{\alpha \cdot |\mathbf{C}^T\mathbf{T}^1|} \tag{5.8}$$

is not below a given tolerance value ε. The relaxation parameter α can be then applied

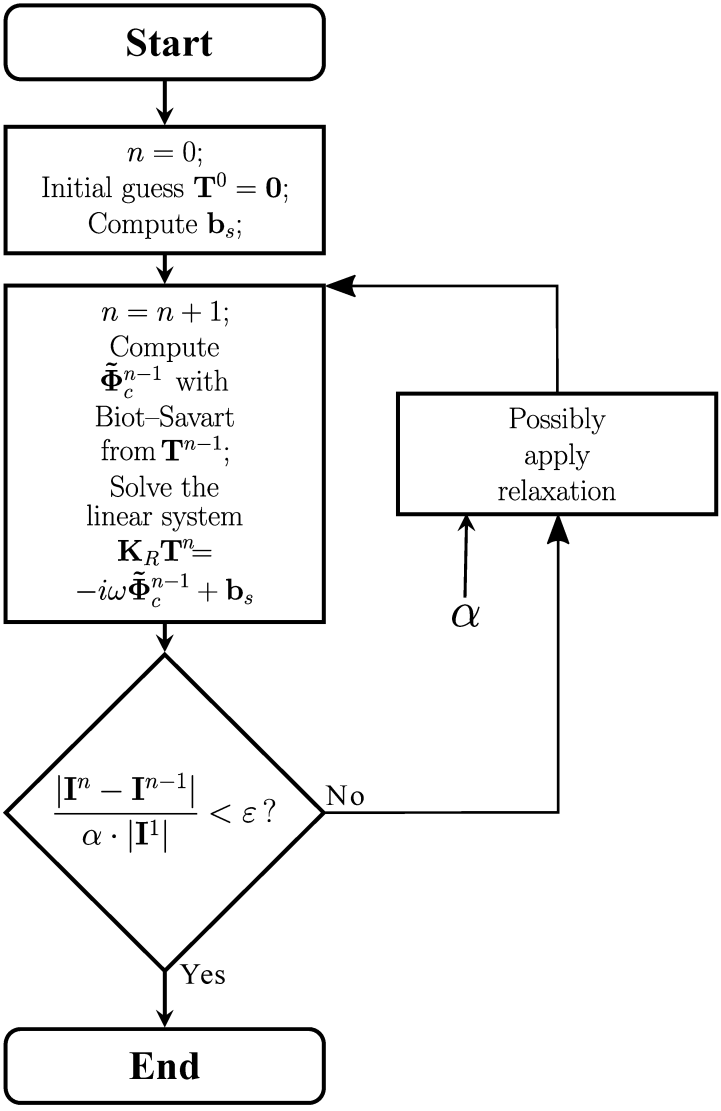

Figure 5.1: Iterative solution of an eddy current problem.

to ensure the method convergence. Initially, this parameter is set to $\alpha = 1$.

Finally, we here notice that the system to be solved at each step of equation (5.6) is a real valued sparse system thus, in addition to the gauged electric vector potential, any other equivalent method can be applied for the computation of $\mathbf{I}^n$, for instance that ones reported in [47].

Non-trivial domains

The proposed algorithm works also for non-simply connected conductors, provided that cohomology basis functions are used inside the electric vector potential formulation. To recast the iterative scheme of equation (5.6), also depicted in figure 5.1, to deal with non-simply connected domains we may start from the full system of equations to be solved when a non-trivial solid conductor is considered. This system, on the cell complex $\mathcal{K}$, reads

$$\begin{aligned} \mathbf{K}\mathbf{T} + (\mathbf{K}\mathbf{H})\,\mathbf{i} &= -i\omega\mathbf{C}^T\tilde{\mathbf{A}}_s, \\ \left(\mathbf{H}^T\mathbf{K}\right)\mathbf{T} + \left(\mathbf{H}^T\mathbf{K}\mathbf{H}\right)\mathbf{i} &= -i\omega\mathbf{H}^T\mathbf{C}^T\tilde{\mathbf{A}}_s. \end{aligned} \tag{5.9}$$

If $\mathbf{K}$ is split, as it has been previously done, $\mathbf{K} = \mathbf{K}_R + i\omega\mathbf{K}_M$ and (5.6) is plugged into (5.9), the following Jacobi iterative scheme is obtained:

$$\begin{aligned} \mathbf{K}_R\mathbf{T}^n + (\mathbf{K}_R\mathbf{H})\,\mathbf{i}^n &= -i\omega\left(\tilde{\mathbf{\Phi}}_c^{\,n-1} + \mathbf{C}^T\tilde{\mathbf{A}}_s\right), \\ \left(\mathbf{H}^T\mathbf{K}_R\right)\mathbf{T}^n + \left(\mathbf{H}^T\mathbf{K}_R\mathbf{H}\right)\mathbf{i}^n &= -i\omega\mathbf{H}^T\left(\tilde{\mathbf{\Phi}}_c^{\,n-1} + \mathbf{C}^T\tilde{\mathbf{A}}_s\right), \end{aligned} \tag{5.10}$$

which is the sought iterative scheme.

In the prosecution, we will show that the use of lazy generators algorithm for the computation of the representatives of the $H^1(\partial\mathcal{K})$ cohomology generators does not affect the convergence of the method.

5.1.2 Convergence criteria

If we consider an arbitrary linear system in the form

$$\mathbf{A}x = \mathbf{b}, \tag{5.11}$$

we can rewrite the left hand side as $\mathbf{A} = \mathbf{S} + \mathbf{N}$ where $\mathbf{S}$ is an arbitrary sparse matrix and where $\mathbf{N}$ is a dense one. Then, by means of $\mathbf{A}$ partition, the general form of a fixed point iterative scheme can be expressed as

$$\mathbf{S}x^n = -\mathbf{N}x^{n-1} + \mathbf{b} \tag{5.12}$$

where it is possible to express the unknown values at each iterate as

$$x^n = -\mathbf{P}x^{n-1} + \mathbf{S}^{-1}\mathbf{b}, \tag{5.13}$$

where

$$\mathbf{P} = \mathbf{S}^{-1}\mathbf{N}$$

is the *iteration matrix* of the iterative scheme of (5.12). In order to ensure the method convergence, the following inequality

$$\|\mathbf{P}\| < 1 \tag{5.14}$$

has to be verified, where $\|\mathbf{P}\|$ is any matrix norm induced by the metrics.

Yet, it is not always possible to obtain an explicit expression of $\mathbf{P}$ and indeed this is the case since in (5.10) $\mathbf{N}$ matrix is not assembled but it is "hidden" inside the computation of $\tilde{\mathbf{\Phi}}$. For this reason, we take advantage of [75] where eddy currents are solved with an iterative scheme by resorting to a reinterpretation of the $A-V$ formulation as an integral formulation. In that case, it is explained that, for the method to be always convergent, a relaxation parameter has to be applied so that, at each step, the system right hand side (RHS) of equation (5.6) becomes[1]

$$\mathbf{K}_R \mathbf{T}^n = \beta^{n-1}, \tag{5.15}$$

$$\beta^{n-1} = \alpha \left(-i\omega \tilde{\mathbf{\Phi}}_c^{n-1} + \mathbf{b}_s \right) + (1-\alpha) \left(RHS^{n-2} \right). \tag{5.16}$$

More than this, it is also given an upper and lower boundary for the values of α so that

$$0 < \alpha < \frac{1}{1 + \frac{4}{3} \left(\frac{R}{\delta} \right)^4} \tag{5.17}$$

where R is the radius of the smallest ball containing the computational domain and $\delta = \sqrt{\rho/\pi\mu f}$ is the penetration depth. The symbols ρ, μ and f have already been defined.

Consequently, equation (5.17) suggests that $\alpha < 1$ has to be chosen and that the convergence behaviour is influenced by R, f, ρ and μ. Indeed, this is what it is shown in the continuation.

Convergence tests

In order to reconstruct the convergence behaviour of this iterative formulation, we performed different tests on a simple geometry used as test bench. This geometry is represented in figure 5.2 where a conducting solid plate whose dimension are $20 \times 20 \times 2\,\mathrm{mm}$ is shown. Starting from the discretization proposed in the figure, other two grids of the same conductor have been produced, by progressively increasing the grain of the mesh: by splitting each elements along x, y and z direction, three grids formed by 32, 256 and 1048 hexahedral volumes have been considered for these tests. In each test case problem, the conducting plate has been immersed in a uniform external magnetic field directed along z axis: $B_z = 1\,\mathrm{T}$. The other parameters of the problem such as the resistivity ρ, the frequency f and the overall size of the domain R were varied while the magnetic permeability $\mu = \mu_0$ was kept constant. In addition, also the modulation of the relaxation parameter α has been investigated.

As initial test, convergence has been studied by only varying the relaxation parameter α and the frequency f on the three considered meshes. The related results are reported

[1] The application of the relaxation parameter refers to the simply-connected domain case for clarity.

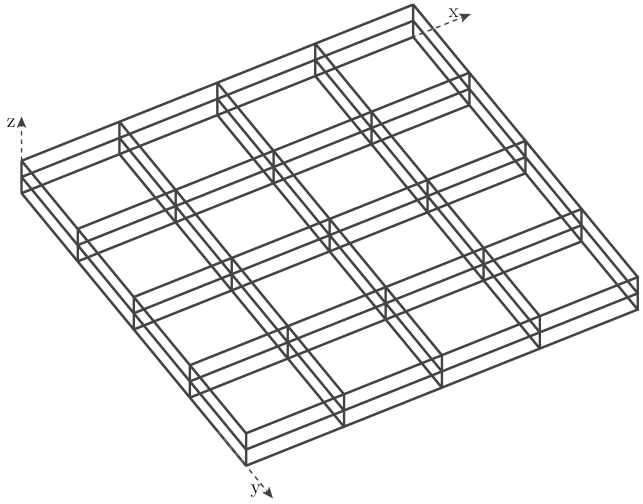

Figure 5.2: The solid conducting plate meshed with the coarsest grid (32 volumes).

in figure 5.3. From the chart, as first aspect, we point out that there is no dependency of the convergence on the mesh grain: for each different mesh, once the frequency or the value of α have been established, almost the same number of iterations is required to reach the same tolerance ε. Differently, as far as the influence of α on the convergence behaviour is concerned, we notice that no over relaxation can be taken into consideration to improve the convergence as recommended in equation (5.17) and differently than what is reported in [73] where values of $\alpha > 1$ are used to accelerate the convergence. This last point suggests that the use of an *integral* formulation, as in [75] and differently from [73] (where a *differental* formulation is applied), changes the spectral radius of the problem matrix thus requiring $\alpha < 1$ to ensure convergence. As far as the frequency is concerned, one more time we obtain results that are in accordance with equation (5.17) since by increasing the frequency of the source field also the number of necessary iterations increases i.e. the value of α has to be diminished in order to have a convergent scheme: in the tests reported in the graph, $\alpha = 0.1$ was chosen to ensure convergence for all the selected frequency values.

In figure 5.4 we can appreciate the effect of α on the current update variation. As expected, the lower is α the higher is the number of iterations required to reach ε thus confirming the role of α in *slowing down* the iteration process: if this effect is not necessary when working either at low frequency or high resistivity thus favouring the choice of $\alpha = 1$, when these parameters become more critical, lower and lower values of the relaxation parameter are necessary for the scheme to be convergent. This trend is also matched by the results displayed in table 5.1 where the plate dimensions were increased (10 times and 50 times, respectively) with respect to the original size previously declared: an increment of R reflects into requiring lower values of α. Similar results have been obtained when the resistivity ρ is varied: the lower is ρ the lower has to be α to ensure the convergence.

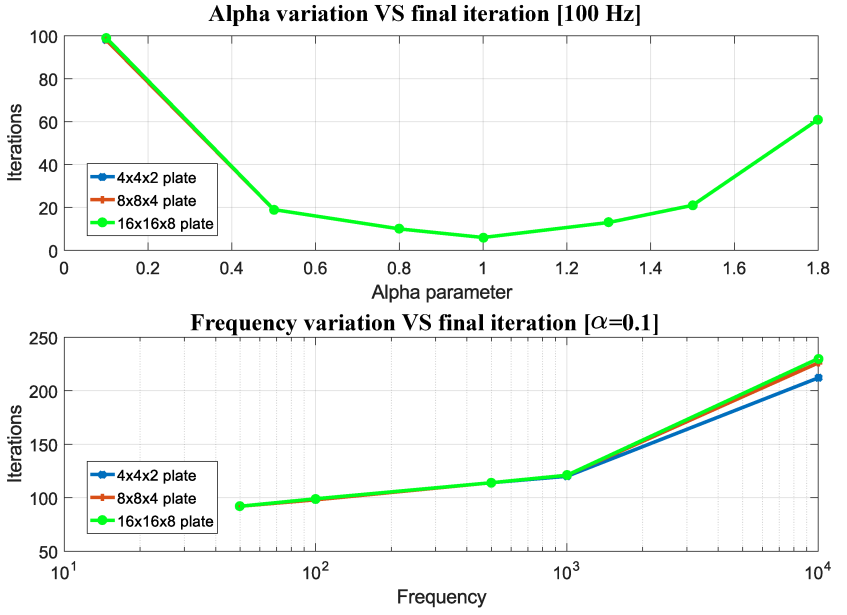

Figure 5.3: Top: required iterations with relaxation parameter α and mesh grain variation. Bottom: Required iterations trend with frequency and mesh grain variation.

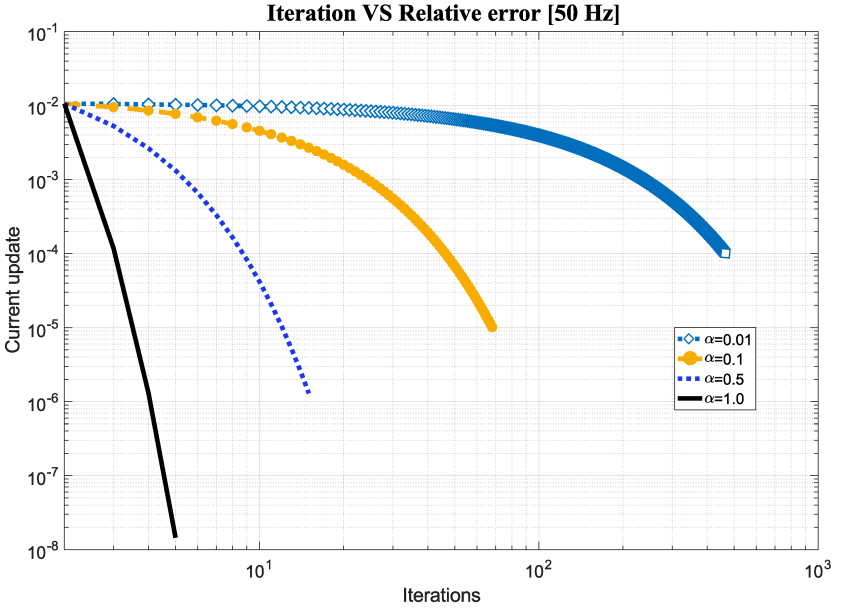

Figure 5.4: Current update trends up to convergence with the relaxation parameter α variation.

Table 5.1: Domain size (R) variation

Domain size	1x	10x	50x
α_{max} value	1	0.5	0.01
Iterations	5	30	964

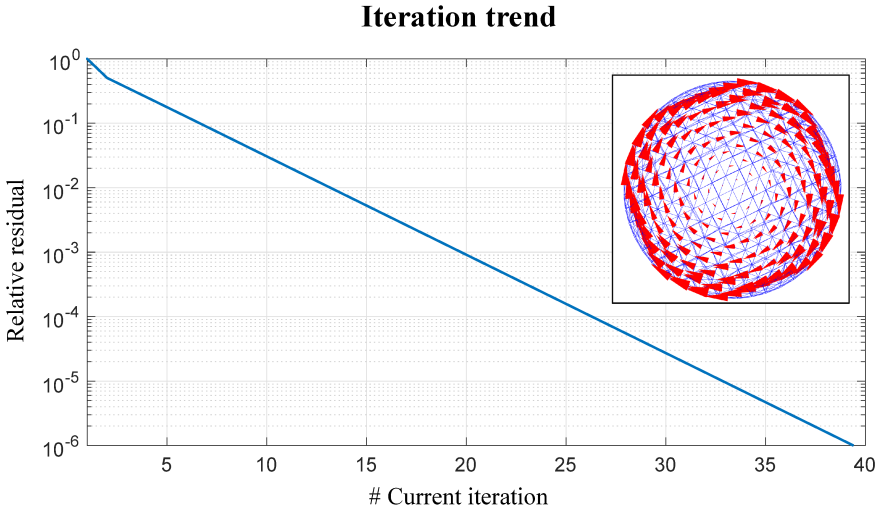

Figure 5.5: Inset: imaginary part of the current density field induced in a solid sphere ($r = 50\,\text{mm}$, $\rho = 0.1\,\mu\Omega\text{m}$) immersed in a uniform sinusoidal magnetic field ($B_z = 1\,\text{T}$, $f = 50\,\text{Hz}$) whose direction is perpendicular to the page. Main graph: convergence trend. The current update (namely "Relative residual") is reported as a function of the current iteration.

5.1.3 Numerical results

In order to test the method accuracy, the iterative scheme has been used to compute eddy currents induced in a solid sphere of radius $r = 50\,\text{mm}$ and resistivity $\rho = 0.1\,\mu\Omega\text{m}$ subject to a vertical sinusoidal magnetic induction field $B_z = 1\,\text{T}$ varying at a frequency $f = 50\,\text{Hz}$. The computational domain consists of a polyhedral mesh (the same as the one of section 4.2.4) constituted by 1840 polyhedral volumes. After that the proper tree-cotree decomposition has been applied, the degrees of freedom of the problem are DOFs = 4329. At the given frequency and resistivity, $\alpha = 0.5$ was chosen as relaxation parameter to achieve the iteration convergence. The present solution, reported in figure 5.5, is in excellent agreement with the one reported in section 4.2.4 which was obtained with the volume integral code. In addition, in figure 5.6 another comparison is presented that considers the induction field computed by means of the approximate solution obtained with this iterative scheme versus the field analytically calculated: once more, the error is comparable with the one obtained by solving the problem with the direct approach (i.e. the volume integral code) thus validating the new method proposed.

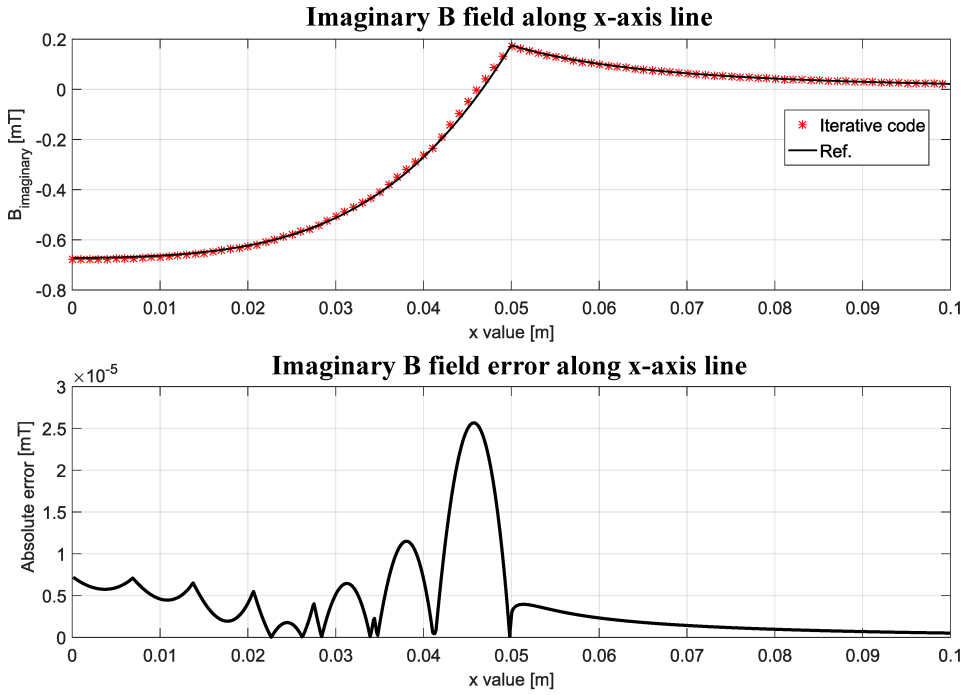

Figure 5.6: Magnetic induction field due to the immaginary part of the eddy currents induced in the sphere and related absolute error.

Last but not least, we present the results obtained by applying this iterative method to a non-trivial domain, shown in figure 5.7. As announced, for the computation of the cohomology basis we took advantage of the lazy generators algorithm which has been applied also to the proposed iterative scheme, in the way explained in equation (5.10). To assess the method accuracy, the results have been compared to the solution obtained by using the volume integral code. With a tolerance set to $\varepsilon = 10^{-6}$, the number of iterations required to obtain the solution is $\mathrm{IT}_{\mathrm{final}} = 3$ with $\alpha = 1$. Moreover, the problem was solved by selecting only the independent generators too (i.e. 4 generators since for this geometry $m = 4$) and the same result has been observed in terms of computed current density (up to the linear solver tolerance) and in terms of final number of iterations required for the convergence.

Tests performed with this iterative formulation in which the whole dense matrix $\mathbf{K}_M$ is not assembled and whose effect is taken into account during the iterative process have shown a poor convergence behaviour. In fact, if on one hand at low frequency and with a mildly high resistivity the number of iterations is small making this iterative scheme appealing, on the other hand, when the frequency exceed some tens of kHz in a conducting structure made of copper ($\rho \approx 10^{-8}\Omega\mathrm{m}$) the relaxation parameter to be applied to reach convergence becomes particularly small ($\alpha < 0.01$) and the consequent iterations number is prohibitive for practical applications. For this reason, in the next

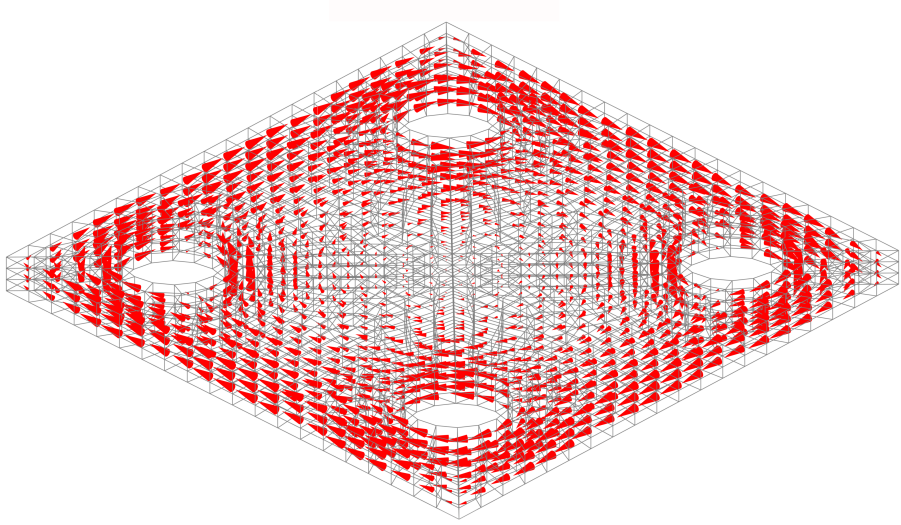

Figure 5.7: Imaginary part of the induced currents in a non-trivial conducting structure (a plate with 4 holes). Plate size: $100 \times 100 \times 25$ mm of resistivity $\rho = 10\,\mu\Omega$m; source field: uniform vertical magnetic field $B_z = 1\,\mathrm{T}$ at a frequency $f = 50\,\mathrm{Hz}$. The degrees of freedom of the problem are DOFs = 2549.

section we propose an improved version of the present iterative method which will be based on domain decomposition techniques to be applied when studying eddy currents on many distinct conductors.

5.2 A direct-iterative hybrid method

Once more, we propose a method to overcome the issue due to the fully populated magnetic matrix $\mathbf{K}_M$ that characterize integral formulations for eddy currents in the frequency domain. More specifically, the formulation developed in the following aim at providing an additional technique to be used when matrix compression based on low-rank approximation is not enough to reduce the number of unknown. For this reason, we resort to a direct-iterative hybrid approach which can be applied when the computational domain is constituted by Ω_i, $i = 1, ..., N$ disjoint conductors.

The present method, inspired by the approach for multiple-scattering problems proposed in [78], takes advantage of the domain splitting to directly solve eddy currents on each conductors Ω_i. Once the current density in each substructure is available, Biot–Savart law is applied in a similar fashion as the prior iterative method to take into consideration the inductive effects between each disjoint subdomain i.e. the outcome that in the original volume integral formulation is expressed by the induction matrix $\mathbf{K}_M$. This strategy leads to several benefits: on one hand, the computation of eddy currents in each substructure as an independent problem improves the convergence because *local* inductive effects are accurately computed; on the other hand, this approach based on the domain splitting allows to reduce the size of the dense matrix to be assembled

and stored since all the off-diagonal and mutual effects are taken into account on the fly by using Biot–Savart law whose computation, as already said, can be performed in an accurate way and can be parallelized [76]. In addition, dealing with many sub problems, instead of Jacobi iterative scheme, we can rely on Gauss–Seidel scheme whose convergence is faster and, for the problems here studied, always achieved without the need of any relaxation parameter. Eventually, to minimize the iterations number, Krylov subspace techniques are considered as an additional tool to be used when Gauss–Seidel convergence slows down.

About this last point, in the continuation we compare different iterative methods based on Krylov subspace in order to identify the best method to obtain robustness and fast convergence[2] and we show that the use of the Generalized Minimum Residual algorithm (GMRES) [79] can fulfil this purpose.

As final remark, we point out that, for the sake of generality, the equations will be given referring to the volume integral formulation suitable for polyhedral grids. Successively, convergence test will be performed by means of the boundary integral method that, in section 4.3.1, is shown to be a particular case of the volume integral one thus being equivalent to it.

5.2.1 Direct-iterative method equations

We now recast the equations of the volume integral code in such a way to obtain the desired novel direct-iterative approach. For the sake of simplicity, the equation of the present section refers to the simply-connected domain case given that the formulation can be extended to non-trivial domains by following the reasoning of chapter 4.

As already said, an eddy current problem written in the discrete geometrical network previously described reads

$$\mathbf{KT} = \mathbf{b}_s, \tag{5.18}$$

where we recall, for the sake of completeness, that $\mathbf{K} = \mathbf{K}_R + i\omega\mathbf{K}_M$ with $\mathbf{K}_R$ a sparse matrix representing the discrete counterpart of the electric constitutive law and $\mathbf{K}_M$ the fully populated matrix that stems from the magnetic constitutive relation in an integral form expressed via the Biot–Savart law.

We now introduce the whole computational domain Ω_c which we assume to be constituted by $\Omega_i, i \in \{1, 2, ..., N\}$ distinct conductors (or *subdomains*), partitioning Ω_c in such a way that $\Omega_c = \Omega_1 \cup \Omega_2 \cup ... \cup \Omega_N$ and $\Omega_i \cap \{\Omega_c \setminus \Omega_i\} = 0,\ \forall i = 1, .., N$. Under this hypothesis, we can provide the following block-diagonal matrix

$$\mathbf{K}_{SD} = \begin{bmatrix} \mathbf{K}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{K}_N \end{bmatrix} \tag{5.19}$$

in which each block $\mathbf{K}_i = \mathbf{K}_{R_i} + j\omega\mathbf{K}_{M_i}$ expresses Faraday's law written in the DGA framework for the only subdomain Ω_i. With the same reasoning, the DOFs vector $\mathbf{T}$ is partitioned too, in such a way that each block of the whole $\mathbf{T}$ i.e. each collection of

[2] Fast at least with respect to Gauss–Seidel.

edges, can be related to a subdomain Ω_i as $\mathbf{T} = \begin{bmatrix}\mathbf{T}_1^T \, \mathbf{T}_2^T \, \cdots \, \mathbf{T}_N^T\end{bmatrix}^T$, accordingly to the partition of Ω_c previously described.

If we compare equation (5.19) to (5.18), in order to recover the initial system of equations, the off-diagonal terms have to be alternatively express. These terms describe the magnetic coupling between two substructures Ω_i and Ω_j with $i \neq j$ by means of the magnetic constitutive law. For this reason, in a similar way to what has been done in [80] and in the prior iterative scheme described in section 5.1, we use the current density computed in each subdomain thanks to $\mathbf{K}_{SD}$ matrix to express the inductive coupling between two substructure by directly using the magnetic vector potential $\boldsymbol{a}(\boldsymbol{r})$ expression that for a constant current density in a polyhedral volume v can be calculated by means of [61]. More precisely, this approach differs from the one employed in [80] because in this case no equivalent current is computed on any coupling surface thus introducing an approximation of the original problem expressed in (5.18); instead, this mixed direct-iterative approach aim at equivalently express the initial system of equations.

To that end, we start from the closed formula expressing the magnetic vector potential in a discrete framework constituted by arbitrary polyhedra contained in appendix A to provide a mathematical expression of the coupling between two substructures. Let us now recall equation (A.7), that reads

$$A(\boldsymbol{r}) = \frac{\mu_0 J}{8\pi} \sum_{f_i \in \partial v} (\boldsymbol{r}_f - \boldsymbol{r}) \cdot \boldsymbol{n}_f W_f(\boldsymbol{r}), \tag{5.20}$$

where, indeed, $A(\boldsymbol{r})$ is the magnetic vector potential on a point $\boldsymbol{r}$ generated by a uniform current density J inside a polyhedron v. The meaning of the other geometrical symbols is detailed in the correspondent appendix.

If we introduce the geometrical parameter

$$P_{g,v}(\boldsymbol{r}) := \sum_{S_f \in \partial v} (\boldsymbol{r}_f - \boldsymbol{r}) \cdot \boldsymbol{n}_f W_f(\boldsymbol{r})$$

we can then expand the current density J uniform in v by means of equation (4.5) and we consequently obtain

$$A(\boldsymbol{r}) = \frac{\mu_0 P_{g,v}(\boldsymbol{r})}{8\pi |v|} \cdot \sum_{k=1}^{F} I^k \tilde{f}^k \tag{5.21}$$

where $|v|$ is the volume of v, F the total number of faces of the boundary ∂v of v, I^k the current flowing through the kth face and $\tilde{f}^k$ the kth dual edge.

Later, we introduce the incidence matrix $\mathbf{C}_{ih}$ which is defined as the incidence matrix $\mathbf{C}$ reduced to the faces-edges of the subdomains Ω_i and Ω_h. Then, we also define the array $\tilde{\mathbf{\Phi}}_i$ as the flux of the magnetic field across the dual faces $\tilde{f}$ of Ω_i, generated by the array of currents $\mathbf{I}_h$ flowing in Ω_h. Finally, by recalling equations (3.24), (3.19) and (4.27b), an by introducing the magnetic constitutive matrix $\mathbf{M}_{ih}^{\Delta}$ as $\mathbf{M}$ matrix referred to the couple of subdomains (Ω_i, Ω_h),

$$\tilde{\mathbf{\Phi}}_i = \mathbf{C}_{ih}^T \mathbf{M}_{ih}^{\Delta} \mathbf{I}_h = \mathbf{C}_{ih}^T \mathbf{M}_{ih}^{\Delta} \mathbf{C}_{ih} \mathbf{T}_h := \mathbf{K}_{\mathbf{M}_{i,h}^{\Delta}} \mathbf{T}_h \tag{5.22}$$

holds, where $\mathbf{K}_{\mathbf{M}^{\Delta}_{i,h}}$ is the off-diagonal block matrix that expresses the inductive coupling between two subdomains Ω_i and Ω_h.

If we recall the definition of $\tilde{\mathbf{\Phi}} = \mathbf{C}^T \tilde{\mathbf{A}}$, we can then express $\tilde{\mathbf{\Phi}}_i$ as

$$\tilde{\mathbf{\Phi}}_i = \mathbf{C}_{ih}^T \left[\begin{array}{cccc} \tilde{a}_i^1 & \tilde{a}_i^2 & \cdots & \tilde{a}_i^F \end{array} \right]^T$$

where

$$\tilde{a}_i^{\tilde{f}} := \int_{\tilde{f},i} a^h(\boldsymbol{r}) \cdot \hat{\boldsymbol{t}}\, dl_i. \tag{5.23}$$

For the sake of precision, we remark that $\tilde{a}_i^{\tilde{f}}$ stems from the integral of the magnetic vector potential along the dual edge $\tilde{f}$ of the grid of Ω_i subdomain and that F is the total number of faces or dual edges (because of the one-to-one correspondence between $\mathcal{K}$ and $\tilde{\mathcal{K}}$) of Ω_i.

At this point, if equation (5.21) is plugged into (5.23), each $\tilde{a}_i^{\tilde{f}}$ generated by the array of currents $\mathbf{I}_h \in \Omega_h$ can be written as

$$\tilde{a}_i^{\tilde{e}} = \sum_{n}^{V_q} \sum_{k=1}^{F_{v_h^n}} \frac{\mu_0 I_h^k \tilde{f}_h^k}{8\pi |vol_h^n|} \cdot \int_{\tilde{e},i} P_{g,n}(r) dl_i. \tag{5.24}$$

In this last expression, V_q represents the total number of volumes of Ω_h whereas $F_{v_h^n}$ is the number of faces of a given volume $v_h \in \Omega_h$.

Equation (5.24) leads to the definition of the inductive constitutive matrix between two volumes $v_i \in \Omega_i$ and $v_h \in \Omega_h$ as

$$M_{ih}^{kn} = \frac{\mu_0 \tilde{e}_h^k}{8\pi |vol_h^n|} \cdot \int_{\tilde{e},i} P_{g,n}(r) dl_i. \tag{5.25}$$

In conclusion, the full system of equations in (5.18) can be recast as

$$\mathbf{K}_{SD}\mathbf{T} + i\omega \mathbf{K}_{\Delta}\mathbf{T} = \mathbf{b}_s \tag{5.26}$$

with

$$\mathbf{K}_{\Delta} = \begin{bmatrix} \mathbf{0} & \mathbf{K}_{\mathbf{M}^{\Delta}_{1,2}} & \cdots & \mathbf{K}_{\mathbf{M}^{\Delta}_{1,N}} \\ \mathbf{K}_{\mathbf{M}^{\Delta}_{2,1}} & \mathbf{0} & \cdots & \mathbf{K}_{\mathbf{M}^{\Delta}_{2,N}} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{K}_{\mathbf{M}^{\Delta}_{N,1}} & \mathbf{K}_{\mathbf{M}^{\Delta}_{N,2}} & \cdots & \mathbf{0} \end{bmatrix}. \tag{5.27}$$

Equation (5.26), as desired, is characterized by the splitting between a *self* term $\mathbf{K}_{SD}$ expressing eddy currents inside each substructure and a mutual term $\mathbf{K}_{\Delta}$ that takes into consideration the inductive effects between subdomains. As a consequence, this form is suitable for the application of an iterative formulation, as it is displayed in the following.

Before proceeding, we point out that the mutual matrix $\mathbf{K}_{\Delta}$ has not to be computed and stored for practical purposes. In fact, for the iterative schemes proposed in the continuation, we just need to compute the result of the application of $\mathbf{K}_{\Delta}$ to a given $\mathbf{T}$

since $\tilde{\mathbf{\Phi}}_\Delta = \mathbf{K}_\Delta \mathbf{T}$; in other words, we just need a routine that can *directly* compute $\tilde{\mathbf{\Phi}}_\Delta$. If on one hand the computational time increases because of the iterative procedure and the recursive computation of $\tilde{\mathbf{\Phi}}_\Delta$, on the other hand this approach allows a significant reduction of the memory required to compute eddy currents because only $\mathbf{K}_{SD}$ matrix is assembled and stored This reflects into the possibility of increasing the problem size both in terms of dimension of the conductors to be studied and in term of degrees of freedom (DOFs) of the system thus leading to a more accurate solution. Yet, to avoid that the calculation of $\tilde{\mathbf{\Phi}}_\Delta$ becomes the bottleneck of this approach, the computation of Biot–Savart has to be carefully carried out.

5.2.2 Iterative methods comparison

Starting from equation (5.26), different iteration schemes can be employed to find a vector $\mathbf{T}$ that satisfies it. Given that Jacobi iteration usually exhibits a slow convergence, the most natural choice is thus represented by Gauss–Seidel (GS) scheme that, as previously declared, will be compared to Krylov subspace methods. Specifically, we will study the convergence behaviour of Conjugate Gradient Squared (CGS) method, Bi-Conjugate Gradient Stabilized (BiCGSTAB), Transpose Free Quasi Minimal Residual (TFQMR) and GMRES whose descriptions can be found in [77]. We also remark that a Preconditioned Conjugate Gradient (PCG) method cannot be applied since the iteration matrix is neither hermitian or positive definite [81].

Gauss–Seidel

Gauss–Seidel method applied to equation (5.26) reads

$$\mathbf{K}_i \mathbf{T}_i^{n+1} = \mathbf{b}_{s,i} - i\omega \tilde{\mathbf{\Phi}}_i^n, \quad \forall i = 1, ..., N. \tag{5.28}$$

This reflects into finding a $\mathbf{T}_i \in \Omega_i$ of (5.26) by computing $\tilde{\mathbf{\Phi}}_i = \mathbf{K}_\Delta(i,:)\mathbf{T}$ with the most recent values of $\mathbf{T}_j$, $j \neq i$ available on the other subdomains. The DOFs values $\mathbf{T}_i$ for the considered sub-problem related to Ω_i are then updated to be used in the computation of the successive $\mathbf{T}_{i+1}^{n+1}$ in the subdomain Ω_{i+1}.

Krylov subspace methods

The main advantage of Krylov subspace methods is that their use allows to directly compute the whole $\mathbf{T}$ array instead of finding each $\mathbf{T}_i$ on a single Ω_i step by step. Moreover, in the sequel it will be shown that they ensure convergence also when the inductive coupling between subdomains is strong and Gauss–Seidel method requires several iteration before reaching the imposed tolerance. This happens despite the fact that Krylov subspace algorithm require more than one function evaluation per iteration, in a sense that will be soon clarified.

Broadly speaking, an iterative approach based on Krylov subspace techniques just requires a routine that can compute the left hand side value of (5.26) by applying $\mathbf{K}_{SD}$ and $\mathbf{K}_\Delta$ to an arbitrary DOFs vector $\mathbf{T}^*$. More precisely, this routine will perform the matrix-vector product $\mathbf{K}_{SD}\mathbf{T}^*$ and then add the missing contribution $\tilde{\mathbf{\Phi}}_\Delta^* = \mathbf{K}_\Delta \mathbf{T}^*$ by calculating it on the fly by means of Biot–Savart law. Furthermore, we recall that

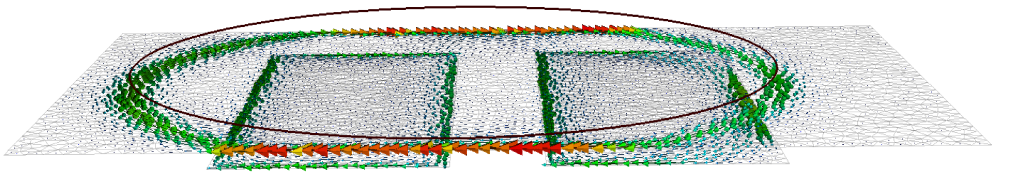

Figure 5.8: Computed $re\{\mathbf{J}\}$ in a conducting domain constituted by three distinct thin plates (3SD). The source of magnetic field is a filamentary wire (red circle above the plates). The dimensions of the major plate are ($l \times w$): $20.0 \times 13.2\,\text{mm}$; the plate thickness is $t = 0.035\,\text{mm}$.

the effectiveness of Krylov subspace methods is influenced by the spectral properties of the iteration matrix (i.e. the left hand side of equation (5.26)); for this reason, for this direct-iterative method, we use $\mathbf{L} = \mathbf{K}_{SD}^{-1}$ as preconditioner for the linear system. By this way, since for the convergence tests that will be presented in the continuation it is chosen that one function call of Krylov method corresponds to N Gauss–Seidel sub-iterations, the computing cost between GS and the other methods can be evenly compared because, due to the preconditioner employed, both for GS scheme and Krylov algorithms the solution of a linear system is necessary. In the GS case the system is solved step by step for each subdomain Ω_i as explained in equation (5.28) whereas for Krylov subspace methods the whole matrix $\mathbf{K}_{SD}$ is used as left hand side of the linear system

$$\mathbf{K}_{SD}\mathbf{T} = \mathbf{f}$$

with $\mathbf{f} = \mathbf{b}_s - i\omega\tilde{\mathbf{\Phi}}_{\Delta}^{*}$. Thus, it turns out that, from the computational point of view, the varying part between GS and Krylov's methods is just represented by the value assigned to the right hand side $\mathbf{f}$ at each step.

Convergence trends comparison

Convergence tests on the proposed direct-iterative hybrid method have been performed by exploiting the boundary integral method. As exposed in section 4.3.1, this last formulation is mathematically equivalent to the volume integral formulation since its equations are derived from that ones of the volume integral code when written for a constant thickness t. Nevertheless, similar convergence tests have been performed also by means of the volume integral formulations leading to similar results.

All the tests have been carried out on the geometries showcased in figures 5.8 and 5.9. The first test geometry is constituted by three thin conducting plates and will thus be denoted as "3SD" geometry, the second one is characterized by eight thin conducting plates thus it will be called "8SD".

As initial test, the convergence behaviour of the iterative schemes previously listed, namely GS, CGS, BICGSTAB, TFQMR and GMRES has been examined on the 3SD geometry at a fixed frequency $f = 10.0\,\text{MHz}$ and for a resistivity $\rho = 1.68 \cdot 10^{-8}\,\Omega\text{m}$ (i.e. assuming that the conducting plates are made of copper). We here notice that in the convergence graphs proposed in the sequel $\mathbf{T}_{ref}$ denotes the reference solution computed with the direct approach offered by the BI formulation of equation (4.36) whereas $\mathbf{T}_k$

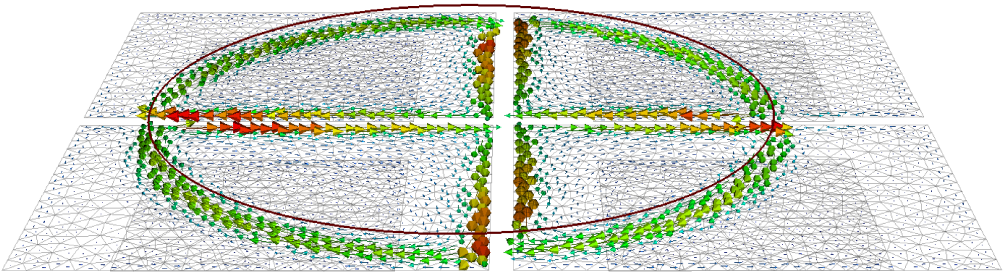

Figure 5.9: Computed $re\{\mathbf{J}\}$ in a conducting domain constituted by eight distinct thin plates (8SD). The source of magnetic field is a filamentary wire (the red circle above the plates).

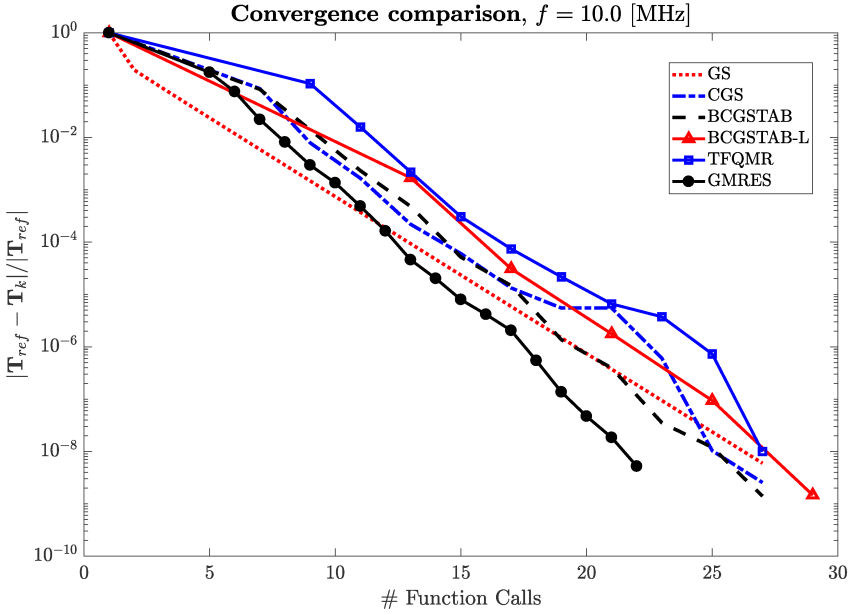

Figure 5.10: Comparison of the convergence behaviour between GS, CGS, BICGSTAB, BICGSTAB(L), TFQMR and GMRES with constant frequency and resistivity.

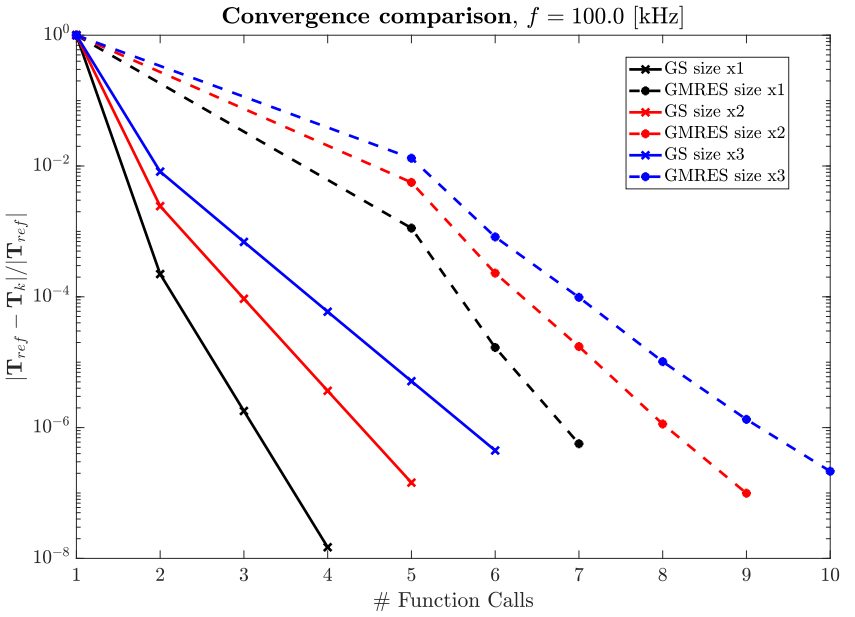

Figure 5.11: Comparison on 3SD geometry (GS vs GMRES). The mesh grain and the thickness of the plates were kept constant, l and w were increased two times (x2) and three times (x3).

is the solution obtained with this direct-iterative hybrid method after k function calls of the considered iterative scheme outlined above.

With the chosen values of frequency and resistivity[3] GMRES shows the best behaviour whereas GS, CGS and BICGSTAB exhibit a similar performance. In addition, we remark that results related to TFQMR and BCGSTAB(L) algorithms are explained by the fact that these two methods requires an higher number of function evaluations per iterate with respect to their competitors such as GMRES or CGS, respectively 4 function evaluations are needed for each iterate of TFQMR and BCGSTAB(L) in place of two function calls for GMRES, CGS and BICGSTAB (refer to [77] for more details). Since this trend repeats also for all the other test conditions the plots related to TFQMR and BCGSTAB(L) will not be reported in the continuation.

As subsequent test, figures 5.11, 5.12 and 5.13 show the outcome of the comparison between GS and GMRES by varying the geometry size, the resistivity and the frequency, respectively i.e. the three influencing parameters of equation (5.17). In the listed plots, we do not report the results obtained with CGS and BICGSTAB schemes for clarity: at low frequency/high resistivity they behave as GMRES (thus GS ensure a better performance) whereas at high frequency/low resistivity they act like GS (thus GMRES turns out to be better in any case in that situation).

With respect to the purely iterative method presented in the previous section, we

[3]This configures as an *high-coupling* condition since the source field variation is high in a low-resistive conducting media.

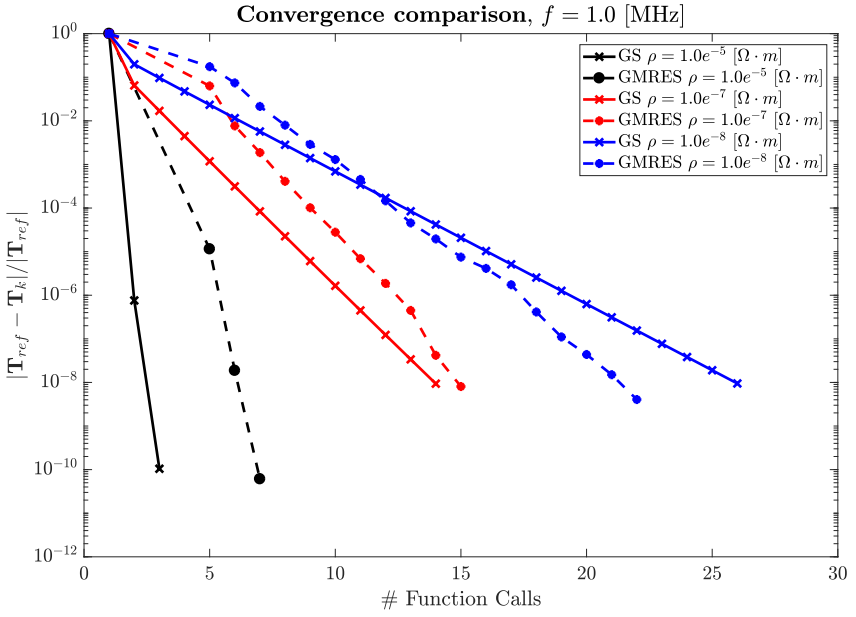

Figure 5.12: Comparison on 3SD geometry (GS vs GMRES): convergence trends by varying the resistivity ρ.

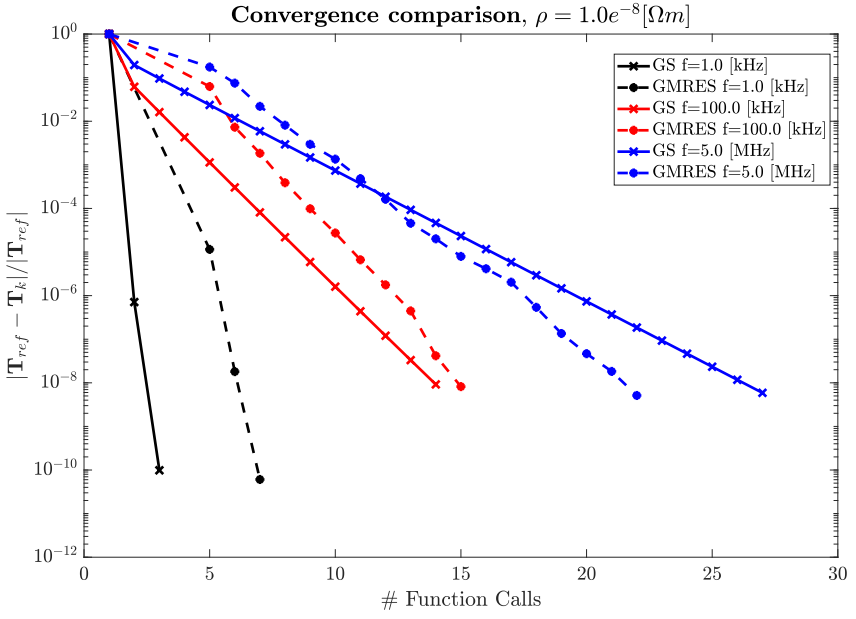

Figure 5.13: Comparison on 3SD geometry (GS vs GMRES): convergence trends by varying the frequency f.

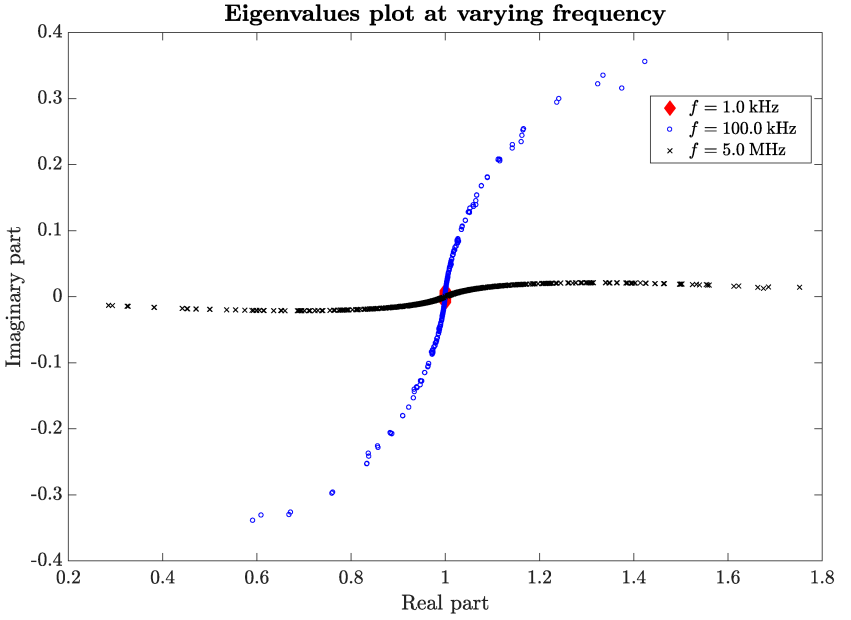

Figure 5.14: Plot of the eigenvalues of the assembled equivalent system when the frequency changes.

can state that this mixed direct-iterative method indeed exhibits a better performance in terms of convergence. First of all, there is no need of any relaxation parameter to be chosen for each different configuration, second, the present method results to be way more stable if we look at its sensitivity for a variation of the geometry size. On the other hand, we observe a similar behaviour when either the frequency or the resistivity changes but, again, the method effectiveness is less influenced by these two parameters thanks to the use of GMRES algorithm that limits the iterations number even when the frequency exceeds some MHz and the resistivity is low. As far as this last point is concerned, reasons of the variation of the convergence behaviour with respect to the frequency, for instance, are well explained by figure 5.14: this figure reports the complex eigenvalues λ of GMRES preconditioned iteration matrix, that has been completely assembled for this test. The graph shows how the frequency acts on λ values that, when f increases, are moved away from the unity i.e. the spectral radius of the matrix increases thus adversely influencing the convergence of the iterative schemes. This explains why under this condition the convergence trend of Gauss–Seidel scheme slows down while GMRES becomes more and more competitive in limiting the iterations number.

Finally, a further test was developed to study the influence of the number of Ω_c partitions N on the convergence. For that goal, we took advantage of 8SD geometry by gathering its eight subdomains into different groups. By this way, three test geometries characterized by $N = 2$, $N = 4$ and $N = 8$ partitions were obtained. In regard to this aspect, the tests performed have shown that the variation of Ω_c partition does not affect the convergence at all: in every test condition, the number of function calls does

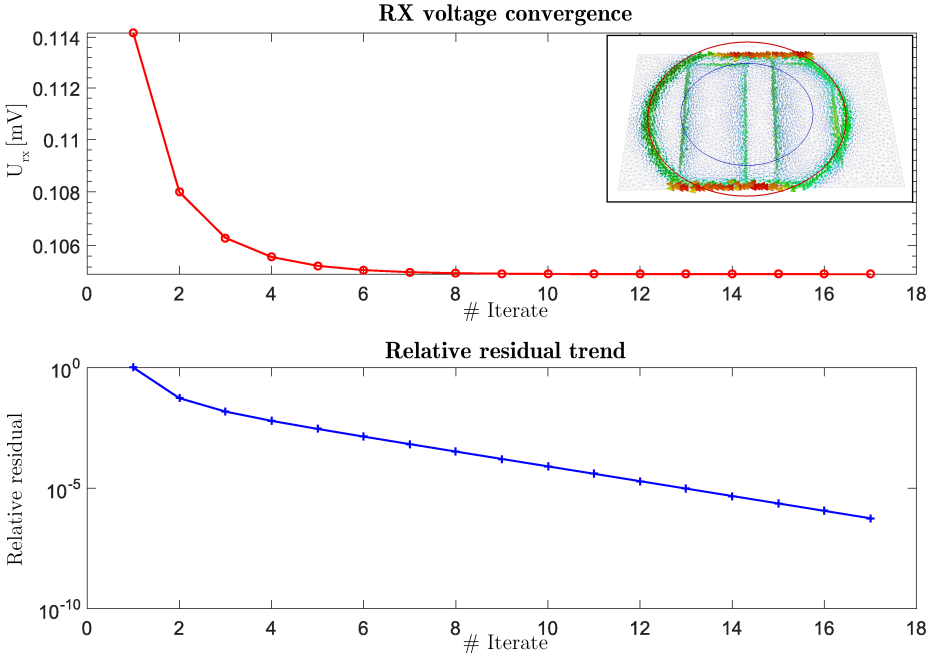

Figure 5.15: Inset: the 3SD geometry setting with the source coil (red circle) and an open-circuit coil (blue circle) in which a voltage U_{rx} is induced. Top: value of the induced voltage in the open-circuit coil after each Gauss–Seidel iteration. Bottom: the relative residual between U_{rx} computed with the present approach and the direct computation of the boundary integral method code used as reference.

not change. The method acts in a similar way even if the mesh grain decrease thus increasing the number of elements of the computational domain: under this condition, the computed current density field is more accurate but the number of function calls is not affected.

Eventually, a more practical problem was solved, by resorting to 2SD geometry, whose results are reported in figure 5.15. The goal of this problem is the computation of the induced voltage on an open-circuit filamentary coil positioned above three conducting plates partially shielding the source field generated by a filamentary wire in which a fixed current flows $I_c = 25\,\mathrm{mA}, f = 1.0\,\mathrm{MHz}$. The graphs show that the solution is achieved within some iterations, moreover we observed that the same accuracy is obtained when computing U_{rx} with the direct-iterative method or when using the BI code.

In conclusion, the proposed direct-iterative hybrid method can be applied in addition to matrix compression techniques when a limitation in the memory occurs due to the fact that with this approach most part of the dense matrix of the volume integral formulation has not to be stored. Furthermore, this formulation was shown to be more effective and robust than the previous one in terms of convergence behaviour. Indeed, it has been observed that when working at frequencies within some tens of kHz Gauss–Seidel iterative scheme exhibits a robust and quite fast convergence without any need of a relaxation parameter whereas when working with high-coupling conditions (i.e. frequency in the order of some MHz in presence of a highly conducting domain) GMRES scheme can be exploited in order to limit the required iterations and achieve convergence faster than GS.

6

Conclusion

In this work, a new volume integral code suitable to deal with non-trivial polyhedral grids has been presented. To that end, some paramount theoretical concepts have been first recalled for a better comprehension of the mathematical basis on which this new numerical model relies. In particular, the author has tried to stress the connection existing between differential p-forms and the *discrete* p-forms which we referred to for the mathematical dissertations here contained. In fact, the choice of detailing all the mathematics by referring to a geometrical framework only without carefully introducing also the concepts related to p-forms based on vectors and co-vectors does not stem from claiming the major correctness of such approach with respect to others, but just from giving to the whole thesis a more logical development.

With the same aim, also the basic topology-related concepts have been exposed in order to be ready-to-use when discussing about the volume integral code gauging, about the application of the proposed code to non simply connected domains and about the delicate aspects underneath the exploitation of cyclic symmetry both on boundary and on volume integral methods.

About this last topic, we here state that the purpose of this work was twofold: on one hand the last findings in regards to cohomology computation related to the integral formulations here presented have been exposed, on the other hand some new tools were proposed that are tailored for the integral formulations considered in order to reduce as much as possible the size of the problem to be addressed without necessarily resorting to matrix compression techniques. In fact, since when integral formulations became appealing to tackle problems in which avoiding the discretization of the conducting domain complement represents a fundamental requirement, the introduction of matrix compression has deeply changed the perspectives of these codes resulting to be more and more effective. However, the author also believes that reconsidering certain features that have always represented a sound tool to limit the computational effort such as the exploitation of cyclic symmetry, could be an additional valid way to achieve an improvement for this category of numerical model especially when, as in our case, some subtle aspects related to the topology of the problem had to be investigated and exploited yet.

Similarly, in this perspective, in chapter 5 the possibilities offered by iterative formulations have been explored as an additional tool to be used beyond matrix compression. We started by investigating the possible issues that can arise when applying such method to the volume integral formulation here proposed to then developed a more effective approach able to avoid the hurdles of a purely iterative scheme thanks to the exploitation of the substructures of the computational domain and thanks to the use of iterative algorithms based on Krylov subspace techniques.

As far as this last aspect is concerned, the author is aware that a more thorough and systematic implementation of the proposed tools represents the next necessary step to make them more appealing for the solution of practical problems. However, in this thesis, the focus was mainly set on studying and improving the method convergence that represents the first essential feature for the method applicability. Yet, some results in regard to the application of integral formulations to practical problems have already been obtained during these three years of doctorate thanks to the development of an automatic numerical tool based on the exploitation of the boundary integral formulation for the optimization of printed circuit boards design for the production of inductive positioning sensors: this theme has not been taken into account in this work because of its lack of connection with respect to the other topics treated and because, to the author opinion, the project is not mature enough for such a purpose even if, so far, a patent titled "Sensor coil optimization" has been accepted by the US patent office [82].

Last but not least, another important topic that is worth mentioning and that is strictly related to this work regards the introduction of new shape functions suitable for polyhedra: the shape functions applied to the present volume integral code are based on the polyhedron splitting into sub-volumes thus representing a non optimal solution in terms of memory-access time to retrieve the required geometrical information to assemble the system matrix. About this, an approach based on new shape functions for tetrahedra has already been proposed by the author in [83] and new possible improvements and extensions have already been undertaken.

A

Closed-form formulas for the computation of Biot–Savart law on polyhedra

We here report, for the sake of completeness, the closed-form formulas of [61] that have been used for the data post processing of the volume integral code and in the iterative formulations of chapter 5. Specifically, we focus our attention on the computation of the magnetic vector potential $\boldsymbol{a}(\boldsymbol{r})$ and of the magnetic induction $\boldsymbol{b}(\boldsymbol{r})$ generated by a given current density $\boldsymbol{j}(\boldsymbol{r'})$ in a volume of the computational domain.

The main idea is rewriting Biot–Savart law in a form suitable for polyhedral meshes i.e. for a discrete representation of a domain in a cell complex $\mathcal{K}$. More than this, differently from numerical integration via Gauss' quadrature rules, these formulas exploit as much as possible vectorial identities in order to obtain *exact* relations at least for a uniform field inside each volume of the mesh, a specification that is coherent with the approximation used for the discretization of constitutive relations used in the VI code beforehand described.

Magnetic vector potential calculation

Biot–Savart law, also used in this work to express the magnetic integral constitutive relation, links $\boldsymbol{a}(\boldsymbol{r}, t)$ to $\boldsymbol{j}(\boldsymbol{r}, t)$. The magnetic vector potential $\boldsymbol{a}(\boldsymbol{r})$ on a point $\boldsymbol{r}$ of a non-magnetic media of permeability μ_0, written for a volume v in which we consider a uniform current density field $\boldsymbol{j}$, reads

$$\boldsymbol{a}(\boldsymbol{r}) = \frac{\mu_0}{4\pi} \int_v \frac{\boldsymbol{j}(\boldsymbol{r'})}{|\boldsymbol{r'} - \boldsymbol{r}|} \, d^3 r' \tag{A.1}$$

where $\boldsymbol{r'}$ is an arbitrary point belonging to v and $d^3 r'$ is an infinitesimal volume of v.

Let us introduce the vector identity

$$\frac{1}{|\boldsymbol{r}-\boldsymbol{r'}|} = \nabla' \cdot \left(\frac{\boldsymbol{r'}-\boldsymbol{r}}{2|\boldsymbol{r'}-\boldsymbol{r}|} \right) \tag{A.2}$$

where $\nabla' \cdot (\bullet)$ stems from the divergence computed by considering $\boldsymbol{r'}$ as variable inside v whereas $\boldsymbol{r}$ can be treated as a constant. By using this last identity and under the hypothesis that $\boldsymbol{j}$ is uniform in the considered volume (and thus, hereafter, it will be denoted as $\boldsymbol{J}$), Biot–Savart law can be written as

$$\boldsymbol{a}(\boldsymbol{r}) = \frac{\mu_0 \boldsymbol{J}}{4\pi} \int_v \frac{1}{|\boldsymbol{r'}-\boldsymbol{r}|}\, d^3r' = \frac{\mu_0 \boldsymbol{J}}{4\pi} \int_v \nabla' \cdot \left(\frac{\boldsymbol{r'}-\boldsymbol{r}}{2|\boldsymbol{r'}-\boldsymbol{r}|} \right) d^3r'. \tag{A.3}$$

If Gauss' theorem is applied

$$\boldsymbol{a}(\boldsymbol{r}) = \frac{\mu_0 \boldsymbol{J}}{8\pi} \oint_{\partial v} \frac{(\boldsymbol{r'}-\boldsymbol{r}) \cdot \boldsymbol{n}}{|\boldsymbol{r'}-\boldsymbol{r}|}\, d^2r' \tag{A.4}$$

holds. In this last relation the symbol $\boldsymbol{n}$ denotes the outgoing normal of an infinitesimal area d^2r whereas ∂v represents the boundary of v.

We now consider the volume v as an arbitrary polyhedron belonging to a discretization of the 3D space encoded into the cell complex $\mathcal{K}$. Under the hypothesis that each face of the polyhedron is a plane surface, we define $\boldsymbol{n}_f$ as *the* outgoing normal vector referred to a face S_f of v (which is constant over the face by its definition). Hence, the following equality holds

$$\int_{S_f} (\boldsymbol{r'}-\boldsymbol{r}) \cdot \boldsymbol{n}\, d^2r' = (\boldsymbol{r}_f - \boldsymbol{r}) \cdot \boldsymbol{n}_f \int_{S_f} d^2r'$$

and the integral over the faces of the polyhedron can be discretized as

$$\boldsymbol{A}(\boldsymbol{r}) = \frac{\mu_0 \boldsymbol{J}}{8\pi} \sum_{f_i \in \partial v} \int_{S_f} \frac{(\boldsymbol{r}_f - \boldsymbol{r}) \cdot \boldsymbol{n_f}}{|\boldsymbol{r'}-\boldsymbol{r}|}\, d^2r' \tag{A.5}$$

where $\boldsymbol{r}_f$ is an arbitrary point of the face f_i, $i = 1, ..., F$ of v whose normal is $\boldsymbol{n}_f$ and $\boldsymbol{A}(\boldsymbol{r})$ stems from the discretization of $\boldsymbol{a}(\boldsymbol{r})$ due to the introduction of the discrete geometrical framework of $\mathcal{K}$.

We then define

$$W_f(\boldsymbol{r}) := \int_{S_f} \frac{1}{|\boldsymbol{r'}-\boldsymbol{r}|}\, d^2r' \tag{A.6}$$

which substituted into equation (A.5) yields

$$\boldsymbol{A}(\boldsymbol{r}) = \frac{\mu_0 \boldsymbol{J}}{8\pi} \sum_{f_i \in \partial v} (\boldsymbol{r}_f - \boldsymbol{r}) \cdot \boldsymbol{n}_f W_f(\boldsymbol{r}). \tag{A.7}$$

If a discrete version of the function W_f can be provided then (A.7) can be used as an exact and closed-form expression for the computation of the magnetic vector potential produced by a uniform current density $\boldsymbol{J}$ inside a polyhedron v of the computational

domain.

W_f computation in a discrete framework

In order to recast W_f in a discrete form, we can resort to the vector identity

$$\frac{1}{|\boldsymbol{r'}-\boldsymbol{r}|} = \boldsymbol{n}_f \cdot \nabla' \times \left(\boldsymbol{n}_f \times \frac{\boldsymbol{r'}-\boldsymbol{r}}{|\boldsymbol{r'}-\boldsymbol{r}|} \right) - \frac{[(\boldsymbol{r'}-\boldsymbol{r})\cdot \boldsymbol{n}_f]^2}{|\boldsymbol{r'}-\boldsymbol{r}|^3}. \tag{A.8}$$

By plugging it into (A.6) and by using Stoke's theorem the subsequent equation is obtained

$$W_f(\boldsymbol{r}) = \oint_{\partial S_f} \boldsymbol{n_f} \times \frac{\boldsymbol{r'}-\boldsymbol{r}}{|\boldsymbol{r'}-\boldsymbol{r}|}\, d^1r' - [(\boldsymbol{r_f}-\boldsymbol{r})\cdot \boldsymbol{n_f}] \int_{S_f} \frac{(\boldsymbol{r'}-\boldsymbol{r})\cdot \boldsymbol{n_f}}{|\boldsymbol{r'}-\boldsymbol{r}|^3}\, d^2r' \tag{A.9}$$

with ∂S_f the boundary of each f_i each one formed by its e_i, $i = 1, .., E$ edges and with d^1r' an infinitesimal length around $\boldsymbol{r'}$. In equation (A.9) the same reasoning used in (A.7) has been applied because, again, the factor $[(\boldsymbol{r'}-\boldsymbol{r})\cdot \boldsymbol{n_f}]$ can be regarded as constant inside a plane face S_f and then, by using an arbitrary point $\boldsymbol{r}_f$ of the face, it can been taken outside the integral over S_f. We here remark that the choice of $\boldsymbol{r}_f$ can be different from the one that led to (A.7) but it has to be coherent with the inner orientation of the E edges of f_i via the screw rule: this can be achieved thanks to the incidence matrix $\mathbf{C}$.

Furthermore, we notice that the term

$$\int_{S_f} \frac{(\boldsymbol{r'}-\boldsymbol{r})\cdot \boldsymbol{n_f}}{|\boldsymbol{r'}-\boldsymbol{r}|^3}\, d^2r'$$

is the solid angle Ω_f seen from $\boldsymbol{r}$ and subtended by an arbitrary face S_f. Then, the unit vector $\boldsymbol{u}_e$ which is tangent to an edge l_e of S_f is introduced and, since we have to integrate along straight edges, the equality

$$\int_{l_e} \boldsymbol{n_f} \times \frac{\boldsymbol{r'}-\boldsymbol{r}}{|\boldsymbol{r'}-\boldsymbol{r}|}\, d^1r' = \boldsymbol{n_f} \times (\boldsymbol{r}_e-\boldsymbol{r})\cdot \boldsymbol{u}_e \int_{l_e} \frac{1}{|\boldsymbol{r'}-\boldsymbol{r}|}\, d^1r' \tag{A.10}$$

holds. By substituting it into (A.9), the following expression for W_f is obtained

$$W_f(\boldsymbol{r}) = \left[\sum_{e_i \in \partial S_f} \boldsymbol{n_f} \times (\boldsymbol{r_e}-\boldsymbol{r})\cdot \boldsymbol{u_e} w_e(\boldsymbol{r}) \right] - [(\boldsymbol{r_f}-\boldsymbol{r})\cdot \boldsymbol{n_f}]\, \Omega_f(\boldsymbol{r}), \tag{A.11}$$

where $w_e((r))$ is defined as

$$w_e(\boldsymbol{r}) := \int_{l_e} \frac{d^1r'}{|\boldsymbol{r'}-\boldsymbol{r}|}. \tag{A.12}$$

The function $w_e(\boldsymbol{r})$ is computed as

$$w_e(\boldsymbol{r}) = \ln \left(\frac{|\boldsymbol{r_2}-\boldsymbol{r}| + |\boldsymbol{r_1}-\boldsymbol{r}| + |\boldsymbol{r_2}-\boldsymbol{r_1}|}{|\boldsymbol{r_2}-\boldsymbol{r}| + |\boldsymbol{r_1}-\boldsymbol{r}| - |\boldsymbol{r_2}-\boldsymbol{r_1}|} \right). \tag{A.13}$$

The points $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$ are the endpoints of a considered edge l_e.

Computation of $\Omega_f(\boldsymbol{r})$

To compute the solid angle seen from the calculation point $\boldsymbol{r}$ and subtended by S_f we resort to the formula for the calculation of the solid angle $\Omega_{\mathcal{T}}$ spanned by a tetrahedron $\mathcal{T}$ whose vertices are $\boldsymbol{r}$, $\boldsymbol{r_1}$, $\boldsymbol{r_2}$, $\boldsymbol{r_3}$. Then, the solid angle Ω_f identified by the polygonal face f_i whose boundary is formed by its $e_i,\ i = 1, ..., E$ edges, is determined by summing the contribution of the E tetrahedra identified by the two endpoints of a given edge e_i, an inner point of the polygonal faces S_f (its barycenter, for example) and by the calculation point $\boldsymbol{r}$.

The angle spanned by $\mathcal{T}$ reads

$$\Omega_{\mathcal{T}}(\boldsymbol{r}) = 2 \arctan \left[\frac{(\boldsymbol{r_1} - \boldsymbol{r}) \cdot (\boldsymbol{r_2} - \boldsymbol{r}) \times (\boldsymbol{r_3} - \boldsymbol{r})}{D} \right] \tag{A.14}$$

where

$$\begin{aligned} D = |\boldsymbol{r_1} - \boldsymbol{r}||\boldsymbol{r_2} - \boldsymbol{r}||\boldsymbol{r_3} - \boldsymbol{r}| + |\boldsymbol{r_3} - \boldsymbol{r}|(\boldsymbol{r_1} - \boldsymbol{r}) \cdot (\boldsymbol{r_2} - \boldsymbol{r}) + \\ + |\boldsymbol{r_2} - \boldsymbol{r}|(\boldsymbol{r_1} - \boldsymbol{r}) \cdot (\boldsymbol{r_3} - \boldsymbol{r}) + |\boldsymbol{r_1} - \boldsymbol{r}|(\boldsymbol{r_2} - \boldsymbol{r}) \cdot (\boldsymbol{r_3} - \boldsymbol{r}). \end{aligned} \tag{A.15}$$

The vertices ordering has to be such that the unit normal vector

$$\boldsymbol{n_T} = \frac{(\boldsymbol{r_2} - \boldsymbol{r_1}) \times (\boldsymbol{r_3} - \boldsymbol{r_1})}{|(\boldsymbol{r_2} - \boldsymbol{r_1}) \times (\boldsymbol{r_3} - \boldsymbol{r_1})|} \tag{A.16}$$

of the face $\{\boldsymbol{r_1}, \boldsymbol{r_2}, \boldsymbol{r_3}\}$ of the tetrahedron $\mathcal{T}$ has the same orientation of $\boldsymbol{n}_f$ of equation (A.11).

Magnetic induction field calculation

The magnetic induction field $\boldsymbol{b}(\boldsymbol{r})$ in a homogeneous non-magnetic media of permeability μ_0 generated by a current density $\boldsymbol{J}$ which is uniform in a volume v writes

$$\boldsymbol{b}(\boldsymbol{r}) = \frac{\mu_0}{4\pi} \int_v \frac{\boldsymbol{J} \times (\boldsymbol{r} - \boldsymbol{r'})}{|\boldsymbol{r} - \boldsymbol{r'}|^3} \, d^3 r'. \tag{A.17}$$

By using the same approach which led to (A.7), we recall the vector identity

$$\frac{(\boldsymbol{r} - \boldsymbol{r'})}{|\boldsymbol{r} - \boldsymbol{r'}|^3} = \nabla' \frac{1}{|\boldsymbol{r} - \boldsymbol{r'}|} \tag{A.18}$$

that plugged into equation (A.17) yields

$$\boldsymbol{b}(\boldsymbol{r}) = \frac{\mu_0 \boldsymbol{J}}{4\pi} \times \int_v \nabla' \frac{1}{|\boldsymbol{r} - \boldsymbol{r'}|} \, d^3 r' = \frac{\mu_0 \boldsymbol{J}}{4\pi} \times \oint_{\partial v} \frac{1}{|\boldsymbol{r} - \boldsymbol{r'}|} \, d^2 r' \tag{A.19}$$

wherein Green's theorem has been applied.

Again, if v is a polyhedron resulting from the space discretization of a conducting

domain Ω_c, whose boundary is formed by its $f_i,\ i = 1, ..., F$ plane faces and since $\boldsymbol{J}$ is uniform in the polyhedron, the discrete expression

$$\boldsymbol{B}(\boldsymbol{r}) = \frac{\mu_0}{4\pi} \sum_{f_i \in \partial v} \boldsymbol{J} \times \boldsymbol{n_f} \int_{S_f} \frac{1}{|\boldsymbol{r} - \boldsymbol{r'}|}\, d^2 r' \tag{A.20}$$

is accomplished.

Finally, by recalling the definition of $W_f(\boldsymbol{r})$ we obtain

$$\boldsymbol{B}(\boldsymbol{r}) = \frac{\mu_0}{4\pi} \sum_{f_i \in \partial v} \boldsymbol{J} \times \boldsymbol{n_f} W_f(\boldsymbol{r}) \tag{A.21}$$

which *exactly* expresses $\boldsymbol{b}(\boldsymbol{r})$ when $\boldsymbol{j}(\boldsymbol{r'})$ is uniform in a polyhedron v of the discrete framework.

How to deal with discontinuities and singularities

As final section, some details are given to be carefully taken into account for a practical implementation of the prior described formulas. More precisely, we now discuss about the presence of discontinuities and singularities and about how to achieve a robust implementation despite these issues.

First, we notice that in the equations previously listed a discontinuity can be found in the computation of the solid angle spanned by a tetrahedron $\mathcal{T}$, namely equation (A.14). In fact, this relation expressing a solid angle should have values between -2π and $+2\pi$ whereas the codomain of $\arctan(\bullet)$ typically spans from $-\pi/2$ to $+\pi/2$. To overcome this limitation, it is just necessary to resort to an inverse tangent function that returns values between $-\pi$ and $+\pi$ (usually called "`atan2`" in many math libraries) to then obtain values of $\Omega_{\mathcal{T}}$ between -2π and $+2\pi$ as desired (because of the "2" factor in (A.14)).

As far as singularities are concerned, they are caused mainly by the zeroing of denominators that can sometimes leads to have indefiniteness of ratios too. Specifically, we are referring to equation (A.12) and equation (A.14) where an incautious choice of $\boldsymbol{r}$ can yield such a situation. For this reason, as suggested in [61], the Euclidean norm of a vector has been re-defined as

$$|\boldsymbol{r'} - \boldsymbol{r}|_\epsilon = \sqrt{|\boldsymbol{r'} - \boldsymbol{r}| + \epsilon^2} \tag{A.22}$$

where ϵ is a constant parameter which has to be small enough with respect to the dimension of the mesh grain. If this hypothesis holds, then (A.22) introduces a negligible error as long as the computation of $\boldsymbol{A}(\boldsymbol{r})$ and $\boldsymbol{B}(\boldsymbol{r})$ is performed for a point $\boldsymbol{r}$ whose distance from the closest edge of the mesh is greater than ϵ. In other words, this means that (A.22) becomes inaccurate for a point $\boldsymbol{r}$ which is sited inside a cylindrical volume of radius equal to ϵ built around the mesh edge on which we are considering $\boldsymbol{r'}$.

B

(Co)homology computation code and algorithms

In this appendix, we report the implementation of the (co)homology routines described in [47] pp. 105–108 as MATLAB® functions. The most remarkable feature of this algorithm which aims at reducing the computational time whenever many generators are to be found, can be identified with the enhanced procedure to retrieve the cocycles used in `lambda_retr()` function. Indeed, instead of building each generators starting from a free edge and then moving step by step along the dual tree thanks to a breadth first search as done in [46], the new routine `BFSdistance()` is separately launched in advance in order to compute the distance field of the nodes of the dual tree graph with respect to the common root. This distance field is defined as the number of dual edges connecting a tree node to the root. Once the distance field *Dist* is available and also the dual tree nodes connections are provided in a proper *Parent* array wherein for each dual tree node the unique ID of that node predecessor is written, each generator is retrieved by going back along the field distance from a free edge not belonging either to the primal or the dual tree until a common node is not reached.

Figure B.1 depicts the main steps above described: with respect to the discussion of chapter 4 this image represents an expansion of figure 4.7 in between snapshots b) and c).

Hereinafter, each section of the code is reported: the pseudocode extracted from [47] is recalled first, then its implementation is shown using as a starting point the main routine of algorithm 1 in chapter 4. Successively, the details of the algorithm subroutines are given.

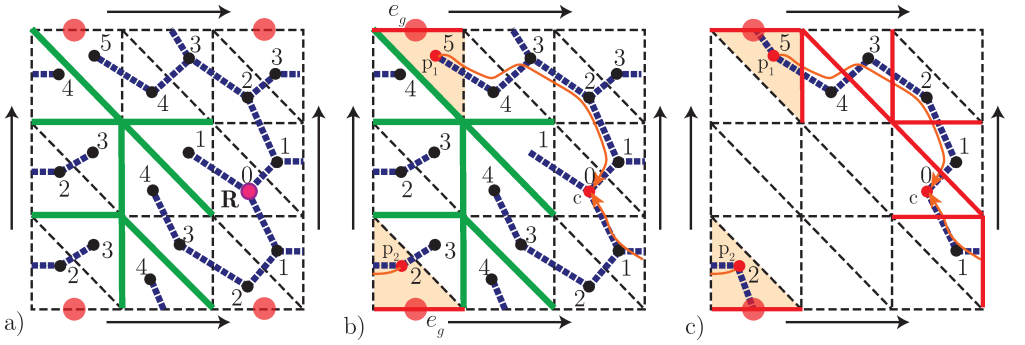

Figure B.1: a) The distance field is computed starting from the root node R of the dual tree. b) One of the free edges e_g is chosen and the cycle is built starting from the dual tree nodes p_1, p_2 whose faces have e_g as common edge. c) The algorithm moves along the dual tree edges until a common node c that closes the cycle is not reached; primal edges (in red) whose dual is the retrieved cocycle are selected and the support of the generator is stored in a column of $\mathbf{H}$.

Main

Algorithm 1 $H^1(\partial\mathcal{K})$ basis generation

Input: $\mathbf{C}_b, \mathbf{G}_b$
Output: $\mathbf{H}^1, ..., \mathbf{H}^{2m}$ generators

procedure GeneratorsRetrieval
  PrimalTree ← *FindPrimalTree*($\mathbf{G}_b$);
  DualTree ← *FindDualTree*($\mathbf{C}_b$, *PrimalTree*);
  [*Dist*, *Parent*] ← *FindBFSdistance*($\mathbf{C}_b$, *DualTree*);
  $g \leftarrow 0$
  for each edge e_g of $\partial\mathcal{K}$ **do**
    if *PrimalTree*(e_g)=*false* and *DualTree*(e_g)=*false* **then**
      $g \leftarrow g + 1$
      $\mathbf{H}^g$ ← *FindCocycleFast*($\mathbf{C}_b$, *DualTree*, e_g);
  return $\mathbf{H}^1, ..., \mathbf{H}^{2m}$

```
function [ Lambda_i ] = H1_generation( C,G,bcond )
%%% H1_generation computes the cohomology basis
%   C,G: incidence matrices; bcond: boundary edges flag array

disp('Cocycle construction')
tic
%% incidence matrices reduction
[edg_ext]=find(bcond==1); %external edges finder

Cb=C(:,edg_ext);
Gb=G(edg_ext,:);

%% Primal and dual indexes retrivial and reduction
%**********************PRIMAL ELEMENTS*********************************
%number of boundary edges
numE=size(Gb,1);
numN=size(Gb,2);

%research of actual boundary nodes (extraction from overall amount of
    nodes of Gb (columns))
node_ind=find(any(Gb)); %node indexes at the boundary
numN_b=numel(node_ind); %nodes number at boundary

%************************DUAL ELEMENTS*********************************
%number of boundary dual edges and nodes[~faces] (TOTAL NUM. INNER AND
    EXTERNAL)
numDN=size(Cb,1);
numDE=size(Cb,2);

%research of actual boundary dual nodes-faces (extraction from overall
    amount of faces of Cb (rows))
punt=sum(abs(C),2)-sum(abs(Cb),2); %if punt(j)=0 then j-th face belongs to
     the boundary

%indexes and quantity of dual nodes i.e. numb and indexes of boundary
    faces
Dnode_ind=find(punt==0);
numDN_b=numel(Dnode_ind);

%% tree retrivial
[ primalT,ccP ] = primaltree(Gb,numE,numN,node_ind,numN_b);

%% dual tree retrivial
[ dualT,ccD ] = dualtree(Cb,primalT,numDN,numDE,Dnode_ind,numDN_b );
%% compute BFS distance from dual tree root node
[ distT,parentT ] = BFSdistance( dualT,Cb,numDN,Dnode_ind,numDN_b );
```

```
%% COCYCLES RETRIVIAL
c=0;
Lambda_i=zeros(numDE,1);
for E=1:numE
        if primalT(E)==false && dualT(E)==false
            c=c+1;
            [lambda] = lambda_retr( Cb,E,distT,parentT,numDE,Dnode_ind );
            Lambda_i(:,c)=lambda;
        end
end
toc
end
```

Find primal tree

Algorithm 2 FindPrimalTree

Input: $\mathbf{G}_b$
Output: $PrimalTree$
procedure FindPrimalTree
initialize vector $PrimalTree$ to $false$ for each edge of $\partial\mathcal{K}$;
initialize vector cc to $false$ for each node of $\partial\mathcal{K}$;
while true **do**
node $R \leftarrow 0$;
for each node R in $\partial\mathcal{K}$ **do**
if $cc(R) = false$ **then**
$break$;
if $R = 0$ **then**
$break$;
$cc(R) \leftarrow true$; initialize queue Q;
for each edge E in $\partial\mathcal{K}$ incidental to R found with $\mathbf{G}_b(:, R)$ **do**
$enqueue(Q, E)$;
while queue Q is not empty **do**
edge $E \leftarrow dequeue(Q)$;
find the nodes N_1 and N_2 boundary of E;
if $cc(N_1) = false$ or $cc(N_2) = false$ **then**
if $cc(N_1) = false$ **then**
$N \leftarrow N_1$;
else
$N \leftarrow N_2$;
$PrimalTree(E) \leftarrow true$;
$cc(N) \leftarrow true$;
for each edge E in $\partial\mathcal{K}$ incidental to N found with $\mathbf{G}_b(:, N)$ **do**
$enqueue(Q, E)$;
return $PrimalTree$;

```
function [ primalT,ccP ] = primaltree( Gb,numE,numN,node_ind,numN_b )
%%% The function builds the primal tree of a graph K starting from G
    matrix

% vectors of nodes and tree inizialization (0=false,1=true)
ccP=false(numN,1);
primalT=false(numE,1);

for ii=1:numN_b
  nn=node_ind(ii);
  if ccP(nn)==false;
      ccP(nn)= true;
```

```
      [primalT,ccP]=BFS_tree(nn,ccP,primalT,numN_b,Gb);
  end
end

end
```

```
function [primalT,ccP]=BFS_tree(nn,ccP,primalT,numN_b,Gb)
%%% BFS function contains the algorithm for the primaltree retrivial

adj1=find(Gb(:,nn)); %edges incident with node nn

% queue initializing
lung=10*numN_b;
[ Q,head,tail ] = inizQ( lung );

% enqueueing adiacent edges
for qq=1:numel(adj1)
  elem=adj1(qq);
  [ Q,tail ] = enqueue( Q, tail,elem );
end
while tail~=head %do while Q is not empty

    [ Q, elem,head ] = dequeue( Q,head );
    edgeE=elem; %extracting one edge from Q tail
    endpt=find(Gb(edgeE,:));
    N1=endpt(1);
    N2=endpt(2);
    if ccP(N1)==false || ccP(N2)==false
        if ccP(N1)==false
            N=N1;
        else
            N=N2;
        end

        primalT(edgeE)=true;
        ccP(N)=true;
        adj2=find(Gb(:,N));
        for qq=1:numel(adj2)
            elem=adj2(qq);
            [ Q,tail ] = enqueue( Q, tail,elem );
        end
    end
end
end
```

Find dual tree

Algorithm 3 FindDualTree

Input: $\mathbf{C}_b$, $PrimalTree$
Output: $DualTree$

procedure FindDualTree
initialize vector $DualTree$ to $false$ for each edge of $\partial\mathcal{K}$;
initialize vector cc to $false$ for each face of $\partial\mathcal{K}$;
while true **do**
face $F \leftarrow 0$;
for each face F in $\partial\mathcal{K}$ **do**
if $cc(F) = false$ **then**
$break$;
if $F = 0$ **then**
$break$;
$cc(F) \leftarrow true$; initialize queue Q;
for each dual edge $D_b(E)$ in $\partial\mathcal{K}$ incidental to dual node $D_b(F)$ found with $\mathbf{C}_b$ and such that $PrimalTree(E) = false$ **do**
$enqueue(Q, E)$;
while queue Q is not empty **do**
edge $E \leftarrow dequeue(Q)$;
find faces F_1 and F_2 such that $D_b(F_1)$ and $D_b(F_2)$ are the boundary of $D_b(E)$;
if $cc(F_1) = false$ or $cc(F_2) = false$ **then**
if $cc(F_1) = false$ **then**
$F \leftarrow F_1$;
else
$F \leftarrow F_2$;
$DualTree(E) \leftarrow true$;
$cc(F) \leftarrow true$;
for each dual edge $D_b(E)$ in $\partial\mathcal{K}$ incidental to dual node $D_b(F)$ found with $\mathbf{C}_b$ and such that $PrimalTree(E) = false$ **do**
$enqueue(Q, E)$;
return $DualTree$;

```
function [dualT,ccD] = dualtree(Cb,primalT,numDN,numDE,Dnode_ind,numDN_b)
%%% The function builds the dual tree of a graph K starting from C matrix

% vectors of Dual nodes and Dual tree inizialization (0=false,1=true)
ccD=false(numDN,1);
dualT=false(numDE,1);

%% dual tree retrieval
for ii=1:numDN_b
```

```
  nn=Dnode_ind(ii);
  if ccD(nn)==false;
      ccD(nn)= true;
      [dualT,ccD]=BFS_dualtree(nn,ccD,primalT,dualT,numDN_b,Dnode_ind,Cb);
  end
end

end
```

```
function [dualT,ccD]=BFS_dualtree(nn,ccD,primalT,dualT,numDN_b,Dnode_ind,
    Cb)
%%% BFS_dualtree() contains the algorithm for the dualtree retrivial

adj1=find(Cb(nn,:)); %Dual edges incident to dual node nn==>finding edges
    next to face nn

% queue initializing
lung=10*numDN_b;
[ Q,head,tail ] = inizQ( lung );

% enqueueing adiacent Dual edges
for qq=1:numel(adj1)
    tt=adj1(qq);
    if primalT(tt)==false;
        elem=tt;
        [ Q,tail ] = enqueue( Q, tail,elem );
    end
end

while tail~=head %do while Q is not empty

    [ Q, elem,head ] = dequeue( Q,head );
    edgeE=elem; %dequeuing the tail element from Q

    Cb_r=Cb(Dnode_ind,:);
    endpt=find(Cb_r(:,edgeE));

    DN1=Dnode_ind(endpt(1));
    DN2=Dnode_ind(endpt(2));

    if ccD(DN1)==false || ccD(DN2)==false
        if ccD(DN1)==false
            DN=DN1;
        else
            DN=DN2;
        end
```

```
        dualT(edgeE)=true;
        ccD(DN)=true;
        adj2=find(Cb(DN,:));
        for qq=1:numel(adj2)
            tt2=adj2(qq);
            if primalT(tt2)==false;
                elem=tt2;
                [ Q,tail ] = enqueue( Q, tail,elem );
            end
        end
    end

end

end
```

Find BFS distance

Algorithm 4 FindBFSDistance

Input: $\mathbf{C}_b$, $DualTree$
Output: $Dist$, $Parent$

procedure FINDBFSDISTANCE
  initialize vector $Dist$ to ∞ for all faces in $\partial\mathcal{K}$;
  while true **do**
    $T \leftarrow 0$;
    for each face T in $\partial\mathcal{K}$ **do**
      if $Dist(T) = \infty$ **then**
        break;
    if $T = 0$ **then**
      break;
    $Dist(T) \leftarrow 0$; initialize queue Q; $enqueue(Q, T)$;
    while queue Q is not empty **do**
      face $F \leftarrow dequeue(Q)$;
      for faces $G \in \partial\mathcal{K}$ sharing an edge E with F and $DualTree(E) = true$ **do**
        if $Dist(G) = \infty$ **then**
          $Dist(G) \leftarrow Dist(F) + 1$;
          $Parent(G) \leftarrow F$;
          $enqueue(Q, G)$;
  return $Dist$, $Parent$;

```
function [distT,parentT] = BFSdistance(dualT,Cb,numDN,Dnode_ind,numDN_b)
%%% BFS distance computes the dual node distances and the parents of each
    dual node

%dist vector initialization; Dim = global faces dim (size(C,1))
distT=inf(numDN,1);
parentT=zeros(numDN,1);% each position of the array will contain the index
     of the parent
%of the dual node related to the array position.
%If index=0 that node has not to be considered.

for ii=1:numDN_b
    nn=Dnode_ind(ii);
    if distT(nn)==inf;
        distT(nn)=0;
        lung=5*numDN_b;

        % queue initializing
        [ Q,head,tail ] = inizQ( lung );

        %enqueuing nn element
```

```
        elem=nn;
        [ Q,tail ] = enqueue( Q, tail,elem );

        while tail~=head
            [ Q, elem,head ] = dequeue( Q,head );
            faceF=elem;
            Cb_r=Cb(Dnode_ind,:);
            borderF=find(Cb(faceF,:));
            for ll=1:numel(borderF)
                edg=borderF(ll);
                if dualT(edg)==true;
                    bro_r=find(Cb_r(:,edg));

                    %face indexes = global matrix C
                    faceG1=Dnode_ind(bro_r(1));
                    faceG2=Dnode_ind(bro_r(2));
                    if faceG1==faceF;
                        if distT(faceG2)==inf;
                            distT(faceG2)=distT(faceF)+1;
                            parentT(faceG2)=faceF;
                            [ Q,tail ] = enqueue( Q, tail,faceG2 );
                        end
                    else
                        if distT(faceG1)==inf;
                           distT(faceG1)=distT(faceF)+1;
                           parentT(faceG1)=faceF;
                           [ Q,tail ] = enqueue( Q, tail,faceG1 );
                        end
                    end
                end
            end
        end
    end
end

end
```

Find cocycle fast

Algorithm 5 FastCocycleRetrieval

Input: E, $Dist$, $Parent$, $\mathbf{C}_b$
Output: $\mathbf{H}^i$

procedure FindCocycleFast
find faces F_1, F_2 in $\partial\mathcal{K}$ such that $E \in \partial F_1$ and $E \in \partial F_2$;
$N_1 \leftarrow F_1$; $N_2 \leftarrow F_2$; $\mathbf{H}^i(E) \leftarrow 1$;
while true **do**
if $Dist(N_1) < Dist(N_2)$ **then**
$big \leftarrow N_2; small \leftarrow N_1$;
else
$big \leftarrow N_1; small \leftarrow N_2$;
$circulation \leftarrow \mathbf{C}_b(big, :)\, \mathbf{\Lambda}^i(:)$;
find edge e shared by faces big and $Parent(big)$;
$\mathbf{H}^i(e) \leftarrow -circulation\, \mathbf{C}_b(big, e)$;
if $Parent(big) = small$ **then**
break;
$N_1 \leftarrow small; N_2 \leftarrow Parent(big)$;
return $\mathbf{H}^i$;

```
function [ lambda ] = lambda_retr( Cb,E,distT,parentT,numDE,Dnode_ind )
%%% lambda_retr can find fastly the cocycle given the free edge E.
%   distT and parentT are provided by BFSdistance

%lambda initilization
lambda=zeros(numDE,1);
Cb_r=Cb(Dnode_ind,:);
ff=find(Cb_r(:,E)); %find result is related to reduced indexes of Cb_r

%standard indexes tranformation
nn1=Dnode_ind(ff(1)); %node nn=face ff
nn2=Dnode_ind(ff(2));

%cocycle initialization
lambda(E)=1;

while true
    if distT(nn1)<distT(nn2)
        big=nn2;
        small=nn1;
    else
        big=nn1;
        small=nn2;
    end
```

```
    circ=Cb(big,:)*lambda(:);
    ff_match=0;
    ee_riga=find(Cb(big,:));
    pnt=1; % find indexes "ee_riga" pointer
    while ff_match~=parentT(big)
        ee_pnt=ee_riga(pnt);
        colonn=find(Cb_r(:,ee_pnt));
        if Dnode_ind(colonn(1))==big
            ff_match=Dnode_ind(colonn(2));
        else
            ff_match=Dnode_ind(colonn(1));
        end
        ee=ee_pnt;
        pnt=pnt+1;
    end
    lambda(ee)=-circ*Cb(big,ee);

    if parentT(big)==small
         break;
    else
        nn1=small;
        nn2=parentT(big);
    end
end

end
```

Bibliography

[1] A. E. Ruehli, "Equivalent Circuit Models for Three-Dimensional Multiconductor Systems," *IEEE Transactions on Microwave Theory and Techniques*, vol. 22, no. 3, pp. 216–221, March 1974.

[2] R. Albanese, G. Miano, G. Rubinacci, and R. Martone, "A T formulation for 3D finite element eddy current computation," *IEEE Transactions on Magnetics*, vol. 21, no. 6, pp. 2299–2302, 1985.

[3] R. Albanese and G. Rubinacci, "Finite Element Methods for the Solution of 3D Eddy Current Problems," *Advances in Imaging and Electron Physics*, 1997.

[4] L. Kettunen and L. R. Turner, "A volume integral formulation for nonlinear magnetostatics and eddy currents using edge elements," *IEEE Transactions on Magnetics*, vol. 28, no. 2, pp. 1639–1642, March 1992.

[5] J. Siau, G. Meunier, O. Chadebec, J. Guichon, and R. Perrin-Bit, "Volume integral formulation using face elements for electromagnetic problem considering conductors and dielectrics," *IEEE Transactions on Electromagnetic Compatibility*, vol. 58, no. 5, pp. 1587–1594, October 2016.

[6] M. Kamon, M. J. Tsuk, and J. K. White, "Fasthenry: a multipole-accelerated 3-D inductance extraction program," *IEEE Transactions on Microwave Theory and Techniques*, vol. 42, no. 9, pp. 1750–1758, September 1994.

[7] G. Rubinacci, A. Tamburrino, S. Ventre, and F. Villone, "A fast 3-D multipole method for eddy-current computation," *IEEE Transactions on Magnetics*, vol. 40, no. 2, pp. 1290–1293, March 2004.

[8] S. Kurz, O. Rain, and S. Rjasanow, "The adaptive cross-approximation technique for the 3d boundary-element method," *IEEE Transactions on Magnetics*, vol. 38, no. 2, pp. 421–424, March 2002.

[9] P. Alotto, P. Bettini, and R. Specogna, "Sparsification of BEM Matrices for Large-Scale Eddy Current Problems," *IEEE Transactions on Magnetics*, vol. 52, no. 3, pp. 3–6, 2016.

[10] W. Hackbusch, "Sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices," *Computing (Vienna/New York)*, vol. 62, no. 2, pp. 89–108, April 1999.

[11] C. W. Trowbridge and J. K. Sykulski, "Some key developments in computational electromagnetics and their attribution," *IEEE Transactions on Magnetics*, vol. 42, no. 4, pp. 503–508, 2006.

[12] G. Kron, "Equivalent Circuit of the Field Equations of Maxwell—I," *Proceedings of the IRE*, vol. 32, no. 5, pp. 289–299, 1944.

[13] E. Tonti, *The Mathematical Structure of Classical and Relativistic Physics*. Birkhäuser-Springer, New York, NY, 2013.

[14] ——, "Why starting from differential equations for computational physics?" *Journal of Computational Physics*, pp. 1290–1260, 2014.

[15] B. Delaunay, "Sur la sphère vide. A la mémoire de Georges Voronoï," *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na*, no. 6, pp. 793–800, 1934.

[16] R. Specogna, "Diagonal discrete hodge operators for simplicial meshes using the signed dual complex," *IEEE Transactions on Magnetics*, vol. 51, no. 3, pp. 1–4, March 2015.

[17] J. R. Munkres, *Elements Of Algebraic Topology*. Westview Press, 1993.

[18] E. Tonti, *On the formal structure of physical theories*, 1975.

[19] H. Cartan, *Formes differentielles: Elements de calcul des variations*. Editions Hermann, 1967.

[20] P. Bamberg and S. Sternberg, *A Course in Mathematics for Students of Physics*. Cambridge University Press, 1988, vol. 1, reprint 1998.

[21] G. A. Deschamps, "Electromagnetics and Differential Forms," *Proceedings of the IEEE*, vol. 69, no. 6, pp. 676–696, 1981.

[22] P. Bamberg and S. Sternberg, *A Course in Mathematics for Students of Physics*. Cambridge University Press, 1990, vol. 2, reprint 1996.

[23] J. Dodziuk, "Combinatorial and continuous hodge theories," *Bulletin of the American Mathematical Society*, 1974.

[24] A. Bossavit, "On the geometry of electromagnetism (4) Maxwell's House," *J. Japan Soc. Appl. Electromagn. & Mech.*, no. 6, pp. 318–326, 1998.

[25] ——, "Computational electromagnetism and geometry: Building a finite-dimensional "Maxwell's house" (2) Network constitutive laws," *J. Japan Soc. Appl. Electromagn. & Mech.*, no. 7, pp. 294–301, 1999.

[26] A. Bossavit and J. Verité, "A mixed fem–biem method to solve 3-D eddy-current problems," *IEEE Transactions on Magnetics*, vol. 18, no. 2, pp. 431–435, March 1982.

[27] H. Whitney, *Geometric Integration Theory*. Princeton University Press, 1957.

[28] A. Bossavit, "Whitney forms: A class of finite elements for three-dimensional computations in electromagnetism," *IEE Proceedings A: Physical Science. Measurement and Instrumentation. Management and Education. Reviews*, vol. 135 pt A, no. 8, pp. 493–500, 1988.

[29] F. Trevisan and L. Kettunen, "Geometric interpretation of discrete approaches to solving magnetostatic problems," *IEEE Transactions on Magnetics*, vol. 40, no. 2 I, pp. 361–365, 2004.

[30] T. Tarhasaari, L. Kettunen, and A. Bossavit, "Some realizations of a discrete hodge operator: a reinterpretation of finite element techniques [for em field analysis]," *IEEE Transactions on Magnetics*, vol. 35, no. 3, pp. 1494–1497, May 1999.

[31] A. J. Zomorodian, *Topology for Computing*. Cambridge University Press, 2005.

[32] S. Suuriniemi, "Homological computation in electromagnetic modeling," Ph.D. dissertation, Tampere University of Technology, 2004.

[33] A. Bossavit, "Computational electromagnetism and geometry: Building a finite-dimensional "Maxwell's house" (4) From degrees of freedom to fields," *J. Japan Soc. Appl. Electromagn. & Mech.*, no. 8, pp. 102–109, 2000.

[34] A. Bossavit and L. Kettunen, "Yee-like schemes on staggered cellular grids: a synthesis between fit and fem approaches," *IEEE Transactions on Magnetics*, vol. 36, no. 4, pp. 861–867, July 2000.

[35] P. Dular, J. Y. Hody, A. Nicolet, A. Genon, and W. Legros, "Mixed finite elements associated with a collection of tetrahedra, hexahedra and prisms," *IEEE Transactions on Magnetics*, vol. 30, no. 5, pp. 2980–2983, September 1994.

[36] L. Codecasa, R. Specogna, and F. Trevisan, "Discrete constitutive equations over hexahedral grids for eddy-current problems," *Computer Modeling in Engineering & Sciences*, vol. 31, no. 3, pp. 129–144, 2008.

[37] ——, "A new set of basis functions for the discrete geometric approach," *Journal of Computational Physics*, vol. 229, no. 19, pp. 7401–7410, 2010.

[38] J. Bonelle and A. Ern, "Analysis of compatible discrete operator schemes for elliptic problems on polyhedral meshes," *ESAIM: Mathematical Modelling and Numerical Analysis*, vol. 48, no. 2, pp. 553–581, January 2014.

[39] J. Bonelle, D. A. D. Pietro, and A. Ern, "Low-order reconstruction operators on polyhedral meshes: application to compatible discrete operator schemes," *Computer Aided Geometric Design*, vol. 35–36, pp. 27–41, 2015.

[40] A. Bossavit, "Computational electromagnetism and geometry: Building a finite-dimensional "Maxwell's house" (3) Convergence," *J. Japan Soc. Appl. Electromagn. & Mech.*, no. 7, pp. 401–408, 1999.

[41] ——, "'Generalized Finite Differences' in Computational Electromagnetics," *Progress In Electromagnetics Research*, vol. 32, pp. 45–64, 2001.

[42] A. Bossavit, "Generating whitney forms of polynomial degree one and higher," *IEEE Transactions on Magnetics*, vol. 38, no. 2, pp. 341–344, March 2002.

[43] L. Codecasa, R. Specogna, and F. Trevisan, "Symmetric positive-definite constitutive matrices for discrete eddy-current problems," *IEEE Transactions on Magnetics*, vol. 43, no. 2, pp. 510–515, February 2007.

[44] L. Codecasa and F. Trevisan, "Piecewise uniform bases and energetic approach for discrete constitutive matrices in electromagnetic problems," *International Journal for Numerical Methods in Engineering*, vol. 65, no. 4, pp. 548–565, 2006.

[45] L. Codecasa, R. Specogna, and F. Trevisan, "A geometric integral formulation for eddy-currents," *International Journal for Numerical Methods in Engineering*, vol. 82, no. 13, pp. 1720–1736, 2010.

[46] P. Dłotko and R. Specogna, "Lazy cohomology generators: A breakthrough in (co)homology computations for CEM," *IEEE Transactions on Magnetics*, vol. 50, no. 2, pp. 577–580, February 2014.

[47] P. Bettini and R. Specogna, "Computation of stationary 3D halo currents in fusion devices with accuracy control," *Journal of Computational Physics*, vol. 273, pp. 100–117, 2014.

[48] D. A. Di Pietro and R. Specogna, "An a posteriori-driven adaptive mixed high-order method with application to electrostatics," *Journal of Computational Physics*, vol. 326, pp. 35–55, 2016.

[49] R. Albanese and G. Rubinacci, "Integral formulation for 3D eddy-current computation using edge elements," *IEE Proceedings A: Physical Science. Measurement and Instrumentation. Management and Education. Reviews*, vol. 135-A, no. 7, pp. 457–462, 1988.

[50] L. Kettunen, K. Forsman, and A. Bossavit, "Formulation of the eddy current problem in multiply connected regions in terms of h," *International Journal for Numerical Methods in Engineering*, vol. 41, no. 5, pp. 935–954, 1998.

[51] L. Kettunen, K. Forsman, and A. Bossavit, "Gauging in whitney spaces," *IEEE Transactions on Magnetics*, vol. 35, no. 3, pp. 1466–1469, May 1999.

[52] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Second Edition.* The MIT Press, 2001, pp. 531–539.

[53] A. A. Rodríguez, E. Bertolazzi, R. Ghiloni, and R. Specogna, "Efficient construction of 2-chains with a prescribed boundary," *SIAM Journal on Numerical Analysis*, vol. 55, no. 3, pp. 1159–1187, January 2017.

[54] C. Geuzaine and J.-F. Remacle, "Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities," *International Journal for Numerical Methods in Engineering*, vol. 79, no. 11, pp. 1309–1331, 2009, code available at http://gmsh.info/.

[55] A. Bossavit, "A rationale for 'edge-elements' in 3-D fields computations," *IEEE Transactions on Magnetics*, vol. 24, no. 1, pp. 74–79, January 1988.

[56] T. Kaczynski, K. Mischaikow, and M. Mrozek, *Computational Homology*. Springer New York, 2004.

[57] G. Rubinacci, A. Tamburrino, and F. Villone, "Circuits/fields coupling and multiply connected domains in integral formulations," *IEEE Transactions on Magnetics*, vol. 38, no. 2, pp. 581–584, March 2002.

[58] R. Hiptmair and J. Ostrowski, "Generators of $H_1(\Gamma_h, \mathbb{Z})$ for triangulated surfaces: Construction and classification," *SIAM J. Comput*, vol. 31, no. 5, pp. 1405–1423, 2002.

[59] G. Rubinacci and A. Tamburrino, "Automatic treatment of multiply connected regions in integral formulations," *IEEE Transactions on Magnetics*, vol. 46, no. 8, pp. 2791–2794, August 2010.

[60] P. Dłotko and R. Specogna, "Physics inspired algorithms for (co)homology computations of three-dimensional combinatorial manifolds with boundary," *Computer Physics Communications*, vol. 184, no. 10, pp. 2257–2266, 2013.

[61] M. Fabbri, "Magnetic flux density and vector potential of linear polyhedral sources," *COMPEL - The International Journal for Computation and Mathematics in Electrical and Electronic Engineering*, vol. 28, no. 6, pp. 1688–1700, 2009.

[62] P. Bettini, L. Marrelli, and R. Specogna, "Calculation of 3-D magnetic fields produced by MHD active control systems in fusion devices," *IEEE Transactions on Magnetics*, vol. 50, no. 2, pp. 45–48, February 2014.

[63] A. Bossavit, "Symmetry, groups, and boundary value problems," *Comput. Methods Appl. Mech. Eng.*, vol. 56, no. 2, pp. 167–215, 1986.

[64] S. Kurz, O. Rain, and S. Rjasanow, "Application of the adaptive cross approximation technique for the coupled BE-FE solution of symmetric electromagnetic problems," *Comput. Mech.*, vol. 32, no. 4, pp. 423–429, 2003.

[65] P. Bettini and R. Specogna, "A boundary integral method for computing eddy currents in thin conductors of arbitrary topology," *IEEE Transactions on Magnetics*, vol. 51, no. 3, March 2015.

[66] A. Kameari, "Transient eddy current analysis on thin conductors with arbitrary connections and shapes," *Journal of Computational Physics*, vol. 42, no. 1, pp. 124–140, 1981.

[67] Z. Ren and A. Razek, "Boundary edge elements and spanning tree technique in three-dimensional electromagnetic field computation," *International Journal for Numerical Methods in Engineering*, vol. 36, no. 17, pp. 2877–2893, 1993.

[68] F. Hantila, I. Ciric, A. Moraru, and M. Maricaru, "Modelling eddy currents in thin shields," *COMPEL*, vol. 28, pp. 964–973, 2009.

[69] K. Itō, *Encyclopedic Dictionary of Mathematics*. The MIT Press, 1993.

[70] B. Kapidani, P. Dłotko, P. Alotto, P. Bettini, and R. Specogna, "Computation of relative 1-cohomology generators from a 1-homology basis for eddy currents boundary integral formulations," *IEEE Transactions on Magnetics*, vol. 52, no. 10, October 2016.

[71] M. Pellikka, S. Suuriniemi, L. Kettunen, and C. Geuzaine, "Homology and cohomology computation in finite element modeling," *SIAM J. Sci. Comp.*, vol. 35, no. 5, pp. B1195—-B1214, October 2013.

[72] T. Dey, J. Sun, and Y. Wang, "Approximating cycles in a shortest basis of the first homology group from point data," *Inverse Probl.*, vol. 27, no. 12, pp. B1195—-B1214, October 2011, http://web.cse.ohio-state.edu/~dey.8/shortloop.html.

[73] T. Takagi, T. Sugiura, K. Miyata, S. Norimatsu, K. Okamura, and K. Miya, "Iterative solution technique for 3-D eddy current analysis using T-method," *IEEE Transactions on Magnetics*, vol. 24, no. 6, pp. 2682–2684, November 1988.

[74] T. Takagi, M. Hashimoto, S. Arita, S. Norimatsu, T. Sugiura, and K. Miya, "Experimental verification of 3D eddy current analysis code using T-method," *IEEE Transactions on Magnetics*, vol. 26, no. 2, pp. 474–477, March 1990.

[75] I. D. Mayergoyz and G. Bedrosian, "Iterative solution of 3D eddy current problems," *IEEE Transactions on Magnetics*, vol. 29, no. 6, pp. 2335–2340, November 1993.

[76] T. Maceina, P. Bettini, G. Manduchi, and M. Passarotto, "Fast and efficient algorithms for computational electromagnetics on gpu architecture," *IEEE Transactions on Nuclear Science*, vol. 64, no. 7, pp. 1983–1987, July 2017.

[77] Y. Saad, *Iterative Methods for Sparse Linear Systems, Second Edition*. Society for Industrial and Applied Mathematics, 2003.

[78] C. Geuzaine, A. Vion, R. Gaignaire, P. Dular, and R. V. Sabariego, "An amplitude finite element formulation for multiple-scattering by a collection of convex obstacles," *IEEE Transactions on Magnetics*, vol. 46, no. 8, pp. 2963–2966, August 2010.

[79] Y. Saad and M. Schultz, "GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Comput.*, vol. 7, no. 3, pp. 856–869, 1986.

[80] G. Rubinacci and F. Villone, "The coupling surface method for the solution of magnetoquasi-static problems," *IEEE Transactions on Magnetics*, vol. 52, no. 3, March 2016.

[81] P. Joly and G. Meurant, "Complex conjugate gradient methods," *Numerical Algorithms*, vol. 4, no. 3, pp. 379–406, 1993.

[82] G. Qama, M. Passarotto, and R. Specogna, "Sensor coil optimization," U.S. Patent 16/024,298, 2019.

[83] M. Passarotto, R. Specogna, and F. Trevisan, "Novel geometrically defined mass matrices for tetrahedral meshes," *IEEE Transactions on Magnetics*, vol. 55, no. 6, pp. 1–4, June 2019.